Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2008**	2. REPORT TYPE	3. DATES COVERED **00-00-2008 to 00-00-2008**
4. TITLE AND SUBTITLE **Performance of Wireless Networks Subject to Constraints and Failures**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Illinois at Urbana-Champaign,Department of Electrical and Computer Engineering,Coordinated Science Laboratory,Urbana,IL,61801**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Recent years have seen a proliferation in the use of wireless multi-hop networks in diverse scenarios ranging from community mesh networks to wireless sensor networks. As wireless networks find application in such wide-ranging arenas and are deployed at large scale, they will increasingly need to operate in the presence of heterogeneous, and often constrained hardware capabilities. Furthermore, fault-tolerant communication algorithms will be re- quired to provide the building blocks for reliable operation in the face of failure and/or disruption. In this dissertation, we have investigated performance and fault-tolerance is- sues in networks of such wireless devices. We have studied two specific problem domains viz., throughput performance in multi-channel wireless networks where devices have het- erogeneous and constrained channel switching capabilities, and feasibility of fault-tolerant broadcast in single channel wireless networks where devices can exhibit Byzantine or crash- stop failure.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**300**	

PERFORMANCE OF WIRELESS NETWORKS SUBJECT TO CONSTRAINTS
AND FAILURES

BY

VARTIKA BHANDARI

B.Tech., Indian Institute of Technology Kanpur, 2002

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2008

Urbana, Illinois

Doctoral Committee:

      Professor Nitin H. Vaidya, Chair
      Professor Tarek Abdelzaher
      Professor P. R. Kumar
      Professor Klara Nahrstedt

Abstract

Recent years have seen a proliferation in the use of wireless multi-hop networks in diverse scenarios ranging from community mesh networks to wireless sensor networks. As wireless networks find application in such wide-ranging arenas and are deployed at large scale, they will increasingly need to operate in the presence of heterogeneous, and often constrained, hardware capabilities. Furthermore, fault-tolerant communication algorithms will be required to provide the building blocks for reliable operation in the face of failure and/or disruption. In this dissertation, we have investigated performance and fault-tolerance issues in networks of such wireless devices. We have studied two specific problem domains, viz., throughput performance in multi-channel wireless networks where devices have heterogeneous and constrained channel switching capabilities, and feasibility of fault-tolerant broadcast in single channel wireless networks where devices can exhibit Byzantine or crash-stop failure.

To Mom and Dad

Acknowledgments

Many influences in my life—family, teachers, friends, alma maters, etc.—have contributed directly or indirectly to this dissertation. This is an endeavour to acknowledge but a few of them.

This dissertation owes much to my advisor Prof. Nitin Vaidya. As his student, I have had the freedom to seek my trajectory, while always having access to his advice. My frequent discussions with him, in which I have appreciated his openness to differing points of view, have played an important role in shaping this research.

I thank Prof. Abdelzaher, Prof. Kumar and Prof. Nahrstedt for serving on my doctoral committee, and for their insightful comments. Many past and current colleagues in the Wireless Networking Group have been very helpful in myriad ways during my graduate school days; I thank all of them, in particular Romit Roy Choudhury, Pradeep Kyasanur, Matthew Miller and Jungmin So.

I acknowledge the Vodafone U.S. Foundation, the National Science Foundation, the U.S. Army Research Office, and the Motorola Center for Communication for financially supporting this research. I also thank Lila Rhoades and other staff at the Coordinated Science Laboratory for help with various administrative matters.

I reserve the final word of acknowledgement for my parents. They laid the foundation for this moment, in my childhood, by fostering in me a spirit of independent thought, and a fascination with the realm of ideas. And their unwavering support, and faith in me, through the years, have made this dissertation—and everything leading up to it—possible.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

i.i.d.   independently and identically distributed

w.h.p.   with high probability

w.l.o.g.   without loss of generality

LL   Link Layer

EWMA   Exponentially Weighted Moving Average

HOL   Head-of-line

Chapter 1

Introduction

Recent years have seen a proliferation in the use of multi-hop wireless networks, in diverse scenarios ranging from community mesh networks, to wireless sensor networks. As these networks are deployed and used at increasingly large scales, economic viability will be an important concern. Moreover, in many cases, the form-factor of the devices may be dictated by the application and deployment scenario.

Given the cost and form-factor considerations, one can anticipate that individual devices may be limited in their functionality, and/or prone to various forms of failure. For instance, even though a large number of frequencies may be available for operation, an individual device's transceiver may only be capable of tuning to a small number of frequencies. Hardware failures may occur with non-negligible probability, making a device unusable. The code on a device may possibly be corrupted or compromised. Despite these occurrences, it is desirable that the network as a whole be capable of tolerating some degree of functional constraints and/or failure on the part of individual nodes, without substantially degrading overall performance.

While sensor networks constitute a major area of interest for such constrained devices, these concerns are by no means exclusively limited to these very low-cost, low complexity devices. One can envision more capable and complex systems being subject to similar problems. In situations where a large number of spectrally-separated frequency bands are available, individual devices may be equipped with re-configurable antennas having limited re-configurability. Sometimes policy issues may enforce constraints, e.g., in cognitive radio networks, presence of active primary users in some frequencies may render them unusable by secondary users. Software bugs in distributed application code may lead to erroneous behavior. Nodes may crash and be rendered nonoperational for varying periods of time. Additionally, in certain scenarios, one may be willing to impose soft functional constraints

if this reduces protocol cost/complexity without significantly affecting performance.

The goal of this research has been to investigate the performance of wireless networks that are subject to various forms of functional constraints or failures. We have focused on two specific problem domains that are very relevant to emerging scenarios for wireless network deployment:

Multi-channel wireless networks where nodes have radio-interfaces with heterogeneous and constrained capabilities Many existing wireless standards, e.g., IEEE 802.11, IEEE 802.15.4, provide for multiple frequency channels. However, most radio transceivers in common use can typically only be active on any one of the available channels at a time. Moreover, each device may only be equipped with a small number of transceivers (often only one). In scenarios with multiple active users, harnessing these multiple channels can lead to substantial performance improvement by increasing the number of feasible concurrent transmissions in the network. This requires appropriate routing and scheduling strategies to distribute the traffic load across interfaces and channels. The complexity is further increased when the devices may be of varying type, cost and capability. Thus, they may have heterogeneous radio capabilities in terms of variable number of available interfaces. Moreover, all interfaces may not be able to switch on all channels, and all channels may not be identical. There has been a substantial body of work on multi-channel wireless networks in the past few years. However, much of it has considered nodes with identical radios, with very limited effort in the direction of handling interfaces with heterogeneous and constrained operational capabilities. With the availability of multiple unlicensed frequency bands for use, it is increasingly relevant to envision devices equipped with radios that can only operate on some part of the total available spectrum. In order to allow a diverse set of devices to operate as part of a single network while still obtaining good performance, sophisticated algorithms for coordination, as well as traffic-load distribution will be required. Developing insight through formal theoretical models is an important precursor in that direction. As part of this dissertation research, we have examined this issue. We have developed theoretical models and formulated results for the same. We have also designed a channel and interface management protocol for multi-channel multi-radio wireless networks, which draws upon some of the insights from our theoretical work, and serves as a proof-of-concept of the potential of developing a general design framework to

2

handle a wide range of heterogeneity in hardware characteristics and capabilities.

Single-channel wireless networks where nodes are prone to Byzantine or crash-stop failures Wireless networks are increasingly finding use in critical scenarios, e.g., industrial monitoring and actuation, first-responder networks, etc.. In these scenarios, the reliability of data communication is of prime importance, and may often be the most relevant metric of interest. Due to fundamental differences in the nature of wired and wireless communication, the design of reliable communication algorithms for wireless networks requires a fresh approach. In particular, the wireless medium is a broadcast medium, i.e., a transmission can be received by many receivers in the vicinity of the transmitter. This characteristic is both an advantage and a disadvantage from the standpoint of reliability. The broadcast characteristic can be exploited to improve reliability by designing algorithms that harness the presence of multiple witnesses to a transmitted message. At the same time, it lays transmissions open to the possibility of collisions and jamming. An influential model for the study of fault-tolerant communication in the past two decades has been the Byzantine fault model, and there is a large body of work that studies Byzantine fault-tolerant communication under different assumptions. Given the distinctive nature of the wireless environment, new and different algorithms are needed for this task. This has led to recent interest in studying this problem in the context of networks with a local broadcast property. As part of this dissertation research, we have examined the potential of exploiting the availability of multiple witnesses to a message transmission in a wireless network,, and have established conditions for the achievability of Byzantine fault-tolerant broadcast in a wireless network setting under certain assumed models. These results provide insight into the potential for leveraging the broadcast nature of the wireless medium to improve reliability.

1.1   Outline

The text of this dissertation can be broadly categorized into two parts, each pertaining to one of the two problem domains discussed above.

Chapters 2-6 pertain to multi-channel wireless networks where devices may have heterogeneous and constrained capabilities. In Chapter 2, we introduce the model for analyzing

performance in the presence of switching constraints, discuss related work, and present some preliminaries. In Chapter 3 and Chapter 4, we present asymptotic connectivity and transport capacity results for adjacent (c, f) assignment and random (c, f) assignment respectively, and also discuss insights obtained from these results. We consider the scheduling implications of heterogeneous channels and radios in networks of realistic scale in Chapter 5, where we present some results on performance of certain maximal scheduler in a multi-channel wireless network. In Chapter 6, we describe the design and evaluation (via simulation) of a channel and interface management protocol for a heterogeneous multi-channel multi-radio network, which draws upon insights from the theoretical results in previous chapters, as well as existing results in the literature. We also discuss interesting directions for future work.

Chapters 7-10 pertain to reliable broadcast in failure-prone wireless networks. In Chapter 7, we introduce the reliable broadcast problem and discuss related work. In Chapter 8, we present results for a locally bounded failure model. In Chapter 9, we describe results for a probabilistic failure model. In Chapter 10, we argue for the need, as well as the potential, to evolve lightweight probabilistic mechanisms for reliable communication that exploit knowledge of physical layer characteristics to achieve reliability, and sketch out a simple algorithm for reliable local broadcast as a proof-of-concept of the same.

We conclude in Chapter 11 by summarizing the contributions of the research performed as part of this dissertation.

General notation and terminology used extensively throughout the text is clarified in Appendix A. Other notation and terminology is introduced prior to first use. Some well-known facts and results that have been used in some of the proofs are compiled together in Appendix E.

Chapter 2

Interface Heterogeneity in a Multi-Channel Wireless Network

Many existing wireless standards provide for multiple frequency channels. For instance, the widely used IEEE 802.11 standard for Wireless Local Area Networks specifies 11 channels (of which 3 are non-overlapping) in the 2.4 GHz ISM band, and 12 channels in the 5 GHz ISM band.[1] The IEEE 802.15.4 standard for Wireless Personal Area Networks also specifies 16 channels in the 2.4 GHz band.

However, typical radio transceivers currently in common use can only be active on any one of the available channels at a time. Moreover, each device may only be equipped with a small number of transceivers. When there are multiple active users in the network, harnessing these multiple channels can lead to substantial performance improvement by increasing the number of feasible concurrent transmissions. This requires appropriate routing and scheduling strategies to distribute the traffic load across interfaces and channels. The complexity is further increased when the devices may be of varying type, cost and capability. Thus, they may have heterogeneous radio capabilities in terms of variable number of available interfaces. Moreover, all interfaces may not be able to switch on all channels, and all channels may not be identical. There has been a substantial body of work on multi-channel wireless networks in the past few years. However, much of it has considered nodes with identical radios, with very limited effort in the direction of handling interfaces with heterogeneous and constrained operational capabilities. Given the availability of multiple frequency bands for unlicensed use, it is increasingly relevant to envision devices equipped with radios that can each only operate on some part of the total available spectrum.

We briefly mention some scenarios of interest:

- The need for low-cost, low-power radio transceivers to be used in inexpensive sensor nodes can give rise to many situations involving constrained switching. Hardware

---

[1] The number of available channels varies in different countries according to local regulations.

complexity (and hence cost), and/or power consumption may be significantly reduced if each node operates only in a small spectral range, and switches between a small subset of adjacent channels (e.g., if the transceiver uses an oscillator with limited tunability). However, if more spectrum is available than a single device can utilize, it may be possible at time of manufacture to lock different devices on to different frequency ranges. Another possible scenario is one in which a node may be equipped with a few simple radios each locked to a single frequency at time of manufacture (a similar scenario is proposed in [93] in the context of untuned radios). Due to the small form factor, at most one of these radios may be able to transmit at a time (receiving simultaneously may or may not possible). Thus, the net effect may be similar to having one transceiver that can switch on a subset of frequencies, but only be active on one at any given time.

- Another recent trend is towards deployment of community mesh networks, where participants in a community each deploy a wireless device at their residence, and the resultant network can be used to extend last mile Internet connectivity, as well as to facilitate peer-to-peer communication within the community. Such networks are typically not likely to have a strongly centralized control, and there exists an element of organic growth, wherein each participant may choose to equip their device with commodity hardware in accordance with their willingness (subject to some minimum capability required for inter-operation). For instance, in a network where all devices are equipped with 802.11b radios, some users may choose to equip their devices with additional 802.11a or 802.11g radios, or may substitute their 802.11b radios with 802.11g radios (802.11g is backward-compatible with 802.11b).

While it may be possible to enforce the condition of uniformity on all devices in a network, and thereby simplify the task of channel coordination, doing so forfeits the possibility of performance gains that may be achieved if heterogeneous capabilities are supported. For instance, in the sensor network scenarios discussed above, one could manufacture all devices to operate on the same small subset of all available frequencies, but that entails leaving the remaining spectrum unutilized. Similarly, in the mesh network scenario, one could use protocols that only support 802.11b, but that would imply a loss of the opportunity to exploit the additional spectrum (in case of 802.11a), or higher transmission rates (in case

6

of 802.11g).

Motivated by such concerns, we study the implications of heterogeneous interface capabilities in a wireless network by studying the asymptotic capacity scaling behavior of a network of devices subject to constraints on the channels they can operate on. While many of the above discussed scenarios involve both heterogeneous interfaces and heterogeneous channel characteristics, we focus our effort on interfaces with limited and heterogeneous channel switching capability, and assume identical channels (we will consider the scheduling implications of channel heterogeneity in Chapter 5).

In this chapter, we introduce some constraint models, describe the network model for our asymptotic capacity results, and discuss related work. We also state and prove some results pertaining to the traffic model.

2.1 Some Models for Channel Switching Constraints

In this section, we describe some switching constraint models that we have formulated and studied. These models assume that each node possesses only one half-duplex interface, which can be active on only one channel at any given time. There are c channels available. All channels are orthogonal, and of equal bandwidth. Each interface can only switch (operate) on f channels out of c, and this set of f channels is dictated by the constraint model. These models assume that $c \geq 2$. When $c = 1$, f can only take one value, viz., 1. This reduces to the case of a single channel for which connectivity and capacity results are already known [42, 43]. Therefore, $c \geq 2$ is the case of interest. Furthermore, the models assume that $2 \leq f \leq c$. In Section 2.5, we explain why $c \geq 2, f = 1$ is disallowed.

2.1.1 Adjacent (c, f) Assignment

In this assignment model, an interface can switch between a set of f contiguous channels where $2 \leq f \leq c$. We assume that the available spectrum is in the form of a single contiguous frequency band, which is divided into c channels numbered $1, 2, ..., c$ in order of increasing frequency. Prior to deployment, each interface is assigned a block location i uniformly at random from $\{1, ..., c - f + 1\}$ and thereafter it can switch to any channel in the set $\{i, ..., i + f - 1\}$. This model is relevant when each individual transceiver has limited tunability, and thus may only switch between a small set of contiguous channels. It is also

7

possible to establish a mapping between this model, and the case of untuned radios [93].

2.1.2 Random (c, f) Assignment

In this assignment model, an interface is assigned a subset of f channels $(2 \leq f \leq c)$ uniformly at random from the set of all possible channel subsets of size f. This model can capture situations where tiny low-cost sensor nodes may be equipped with a transceiver having a bank of f switchable filters (e.g., a design with a filter-bank has been proposed in [86]). This model can also capture scenarios involving small form-factor nodes which are equipped with a few simple radios, each locked to a single random frequency at manufacture time. Due to the small form-factor (leading to a very small separation between the radios), it would typically be infeasible for more than one radio to active simultaneously. Thus, the net effect would be as if each node is equipped with a single radio that can switch over a random subset of channels.

2.2 Asymptotic Capacity Analysis

In their seminal paper [43], Gupta and Kumar introduced the approach of asymptotic capacity analysis to understand the scaling behavior of a wireless network, as the network size increases towards infinity. They defined a quantity–the transport capacity–as a measure of the network's ability to transfer data.

Two network models were considered in [43]: *Arbitrary* networks, and *Random* networks. Of these, we discuss random networks, as this the model we utilize for our results. In the random network case, n nodes are located uniformly at random in the network region. Each node is the source of exactly one flow. It chooses its destination by choosing a point uniformly at random and selecting the node closest to it other than itself. Given this traffic model, the average distance traversed by a flow is of the same order as the network diameter.

In a random-network, the per-flow network capacity is said to be $\Theta(\lambda(n))$ if there exist constants c_1, c_2 such that:

$$\lim_{n \to \infty} \Pr[\text{ throughput } c_1 \lambda(n) \text{ is achievable for each flow }] = 1 \qquad (2.1)$$

$$\lim_{n \to \infty} \Pr[\text{ throughput } c_2 \lambda(n) \text{ is achievable for each flow }] < 1 \qquad (2.2)$$

Two models for interference were defined in [43], viz., the *Protocol Model* and the *Physical Model*. Of these, the Protocol Model for a random network is defined as follows:

All nodes in the network use a common transmission range $r(n)$. A transmission from a node A to a node B is successful if and only if the distance $AB \leq r(n)$ and for any other concurrently active transmitter C, the distance $BC > (1 + \Delta)r(n)$, where Δ is a constant which embodies a *guard-zone* needed to prevent interference.

2.3 Assumed Network Model

We assume a random network with the Protocol model of interference. We now describe the details of the model.

n nodes are located uniformly at random in a unit area torus.[2] All nodes use a common transmission range $r(n)$, which can be appropriately selected.[3]

There are c available channels of bandwidth $\frac{W}{c}$ each. We focus on the case where the total number of available channels $c = O(\log n)$. This is reasonable because, in large scale deployments, the number of nodes will typically be much larger than the number of available channels. Besides, when $c = \omega(\log n)$, there is a substantial capacity degradation even with unconstrained channel switching (as shown in [65]), thus making channelization an increasing liability. Constrained switching can only lead to additional degradation, and potentially unacceptable performance.

We assume the same traffic model as in [43]:

Each node is source of exactly one flow. It chooses a point uniformly at random (we shall henceforth refer to these points as *pseudo-destinations*), and selects the node (other than itself) lying closest to that point as its destination.

---

[2]Since the Protocol Model does not involve an explicit power constraint, the unit area assumption in the Protocol Model can be viewed as simply a normalization of a general area A. Capacity results (in bits/sec) for the unit-area continue to hold for a torus of general area A. Results in bit-meters/sec can be obtained by simply multiplying the unit area results with $\sqrt{A}$. Results regarding critical range for connectivity also simply require a scaling by a factor of $\sqrt{A}$. We also remark that from a physical standpoint, the relevant interpretation is indeed that involving an extended network region (whose area increases as n increases), else as argued in [28], scenarios with ever-increasing network density cease to be physically relevant.

[3]Although we denote it by $r(n)$, the transmission range can potentially be a function not only of n, but also c and f.

2.4 Related Work

Connectivity and Capacity of Wireless Networks   There is a substantial body of prior work on deriving the conditions under which a given network is connected, and conditions for connectivity have been formulated in the context of many different network models. For a unit area network with uniformly distributed node placement, where nodes have a common transmission radius $r(n)$, it was shown in [42] that if $\pi r^2 = \frac{(\log n + b(n))}{n}$, then the network is asymptotically connected with probability 1 iff $b(n) \to \infty$. An alternate model was considered in [123], where nodes deployed uniformly at random may individually modulate their transmission power (and hence range) to ensure that they have a certain number of neighbors. It was proved that each node must be connected to $\Theta(\log n)$ neighbors for asymptotic connectivity with probability 1. The issue of theta-coverage and connectivity was considered in [124]. Another relevant body of work is that on bond percolation in wireless networks, e.g. [34].

In [43], Gupta and Kumar defined the notion of asymptotic transport capacity of a wireless network, and obtained results for the capacity of arbitrary and random networks in a single-channel single-interface scenario for two models of interference, viz., the *Protocol Model* and the *Physical Model*.

For the Protocol Model, they established that in an arbitrary network, the capacity scales as $\Theta(\frac{W}{\sqrt{n}})$ bit-m/s per flow, while in a random network, it scales as $\Theta(\frac{W}{\sqrt{n \log n}})$ bits/s. For the Physical Model, they showed that capacity for random networks is $O(\frac{W}{\sqrt{n}})$ and $\Omega(\frac{W}{\sqrt{n \log n}})$. It was later shown by Franceschetti et al. in [35] that under the Physical Model, a per-flow throughput of $\Omega(\frac{W}{\sqrt{n}})$ can be achieved in a random network. While this may seem as closing the gap in the result of [43], this is not strictly the case, as the model of [35] allows use of different data-rates over different links, but stipulated a common transmission power, whereas in [43], different transmission powers may be used, but all communication requires the same SINR threshold, implying that it occurs at a single common rate (corresponding to a case where only one particular modulation scheme may be available). However, a variation of their construction proves the result for the model of [43], and this is described in [125]. Improved capacity bounds for the Protocol Model were presented in [1]. This work also generalized the notion of *exclusion-regions* to arbitrary shapes that could potentially be used to model interference when using directional antennas.

It was shown in [39] that mobility can increase the capacity of a wireless network, and in fact $\Theta(1)$ throughput per flow is attainable when each node is source and destination for exactly one flow each. The capacity of hybrid networks (those having some infrastructure support in the form of access points) was studied in [76] and [59].

The throughput-delay trade-off was studied in [36], and it was shown that the optimal trade-off is given by $D(n) = \Theta(nT(n))$ where $D(n)$ is delay, and $T(n)$ is throughput. The capacity of ultra-wideband (UWB) networks was studied in [95], and [128].

It was shown in [28] that under the unit area assumption, the Physical Model breaks down when n becomes very large, yielding a singularity, and for a model involving a non-singular attenuation function, the per-flow capacity would be asymptotically limited to $O(\frac{1}{n})$. Franceschetti et al [80], have recently shown that fundamental laws of physics dictate a limit of $O(\frac{1}{\sqrt{n}})$ for per-flow capacity scaling when n nodes are distributed over an area of order n.

In [77], it is shown that for the network/traffic model of [43], and the Protocol Model of interference, the use of network coding only yields a constant factor benefit (this constant factor is a function of the guard zone parameter Δ in the Protocol Model).

A concise presentation of many capacity results is available in [125].

Multi-channel Networks It was also shown in [43] that if the available bandwidth W is split into c channels, with each node having a dedicated interface per channel, the results remain the same as for a single-channel, single-interface scenario. However, an interesting, and fairly common, scenario arises when the number of interfaces m at each node may be smaller than the number of available channels c. This issue was analyzed in [65] and it was shown that the capacity results are a function of the channel-to-interface ratio $\frac{c}{m}$. It was also shown that in the random network case, there are three distinct capacity regions: when $\frac{c}{m} = O(\log n)$, the per-flow capacity is $\frac{W}{\sqrt{n \log n}}$, when $\frac{c}{m} = \Omega(\log n)$ and also $O\left(n\left(\frac{\log \log n}{\log n}\right)^2\right)$, the per flow capacity is $\Theta(W\sqrt{\frac{m}{nc}})$, and when $\frac{c}{m} = \Omega\left(n\left(\frac{\log \log n}{\log n}\right)^2\right)$, the per-flow capacity is $\Theta(\frac{Wm \log \log n}{\log n})$. The issue of interface switching delay was also briefly considered in [65], and it was shown that access to some extra interfaces can allow one to completely mask the switching delay.

In [63], an additional multi-channel scenario is considered where each node has two interfaces that may each be assigned a channel based on traffic patterns, but must thereafter

remain fixed on those. For a permutation routing model, it was shown that the capacity with two fixed interfaces is of the same asymptotic order as that with one fully-switchable interface.

Constraints on Channel Availability and Tuning   Situations in which some channels may be unavailable to some nodes have been considered in some work on cognitive radio. An area-blocking model (with a notion of a *protected radius* around a primary user) is considered in [98], which is similar to the spatially correlated channel assignment model we briefly discuss in Chapter 2. However, the goal of that work is not to determine multi-hop capacity. In [61], a model is considered where channel-sets of neighboring nodes may differ by at most k channels. Some algorithms for node-discovery in such networks are proposed. None of these works has focused on obtaining a formal model of such anticipated spatially correlated constraints for connectivity and capacity analysis.

It was proposed in [93] that extremely inexpensive wireless devices can be manufactured if it is possible to handle untuned radios whose operating frequency may lie randomly within some band. Additional considered possibilities were that each device may have a small number of such untuned radios. The model of [93] involved a source and destination capable of transmitting/receiving on all frequencies concurrently, that are spatially-separated, and must communicate via a back-plane of devices with untuned radios. A random network coding based approach was proposed to relay information between the source-destination pair, and it was shown that $\Theta(c)$ throughput is achievable, where c is the maximum number of disjoint channels possible.

On a related note, constraints that are somewhat similar in spirit are also encountered in optical networks with wavelength-division multiplexing (WDM). In an optical network, all nodes may not be capable of wavelength conversion (see, e.g., [108, 73]). Architectures have been proposed for sparse wavelength conversion [108], such that only a small fraction of nodes have wavelength conversion capability. Architectures where nodes have limited conversion capability have also been proposed [71].

Systems/Architectures with Limited Channel-Switching   A multi-channel multi-hop network architecture has been considered in [99] in which each node has a single transceiver, and nodes have a *quiescent* channel to which they tune when not transmit-

ting. A node wishing to communicate with a destination tunes to its quiescent channel, and transmits the packet to a neighbor whose quiescent channel is the same as that of the destination. Thereafter, the packet proceeds towards the destination on the quiescent channel. This has some similarity to the model and constructions in Section 3.5 and Section 4.6. However, in their case, channel-transitions can happen trivially at the very first hop, since the source node is always capable of tuning to the destination's quiescent channel. In contrast, in our models, interfaces can only switch on some channels, and this needs to be taken into account when routing packets.

2.5   Constraints that Limit Capacity

In this section, we briefly discuss some general constraints on the capacity of the network (for any channel assignment model). Recall that c is the total number of available channels, each channel has bandwidth $\frac{W}{c}$, and f is the number of channels any single interface can operate on. Furthermore, each node is equipped with a single interface.

Source-Destination Constraint for $f = 1$

If $f = 1$, but $c > 1$, then communication between a source and its destination is possible if and only if they are both capable of operating on the same channel. This may not always happen if the channels are assigned in some random manner.

To illustrate, consider the class of switching constraint models where the operational channel-set assigned to individual nodes is i.i.d. Suppose, the probability that i and $dst(i)$ operate on a common channel is at most p. If the traffic model is such that any single node can be the destination of only up to $D(n)$ flows, then we argue thus:

We can obtain at least $\lfloor \frac{n}{2D(n)+1} \rfloor$ source-destination pairs, such that the nodes in each pair are distinct, leading to independent probabilities). The probability that, in at least one of the n source-destination pairs, the source and destination do not operate on the same channel can be lower bounded by the probability that the source and destination in at least one of these distinct pairs do not operate on a common channel. This probability is at least $1 - p^{\lfloor \frac{n}{2D(n)+1} \rfloor} = 1 - e^{-\ln \frac{1}{p} \lfloor \frac{n}{2D(n)+1} \rfloor}$. When $\log\left(\frac{1}{p}\right) = \omega(\frac{2D(n)+1}{n})$, this probability converges to 1, as $n \to \infty$. Hence, the network would have zero capacity. For the adjacent (c, f) and random (c, f) assignments studied in this dissertation, with $c \geq 2, c = O(\log n)$,

this condition indeed holds. Therefore $f = 1$ when $c \geq 2$ yields zero capacity. Therefore, our model definitions (Section 2.1) disallow this possibility.

When $f > 1$, as in the rest of the discussion on asymptotic transport capacity in this dissertation, this constraint does not apply.

Connectivity Constraint

This constraint was first formulated in [43]. Given that each node is a source in the assumed traffic model, if even a single node is isolated (i.e., partitioned from the rest of the network), this would imply that the capacity is trivially zero. Thus, at the very least one requires that no node be isolated. Suppose the necessary condition to avoid isolated nodes is that $r(n) = \Omega(g(n))$. It follows from the interference model that each transmission *occupies* a $\Theta(\frac{1}{r(n)^2})$ area, this limits the spatial re-use in the network to $O(\frac{1}{(g(n))^2})$ concurrent transmissions on any single channel. Besides, each source-destination is separated by average $\Theta(1)$ distance (see [43] for details) and hence average $\Theta(\frac{1}{r(n)})$ hops. This limits the per-flow throughput to $O(\frac{W}{nr(n)})$.

Interference Constraint

It was established in [65] that the per flow capacity is constrained to $O(W\sqrt{\frac{1}{cn}})$ when each node possesses a single interface that is capable of switching to any channel. Since it is always possible to simulate a switching constraint model in a network where interfaces can switch to any channel, any throughput achievable with switching constraints is also achievable in the unconstrained switching case. Therefore, the upper bound of $O(W\sqrt{\frac{1}{cn}})$ also applies to adjacent (c, f)-assignment, and random (c, f)-assignment.

Destination Bottleneck Constraint

This constraint was first articulated in [65]. If the traffic model is such that some node can be the destination of up to $D(n)$ flows, the per-flow throughput is constrained to be $O(\frac{W}{D(n)})$, since the destination must time-share its interface between these $D(n)$ flows.

In the region $c = O(\log n)$, the connectivity constraint turns out to be asymptotically dominant.

2.6 Some Results about the Traffic Model

As stated in Section 2.3, we assume the traffic model of [43]. We now establish some general results pertaining to this traffic model.

Lemma 1. *The number of flows for which any node is the destination is $O(\log n)$ w.h.p.*

Proof. Consider a flow's pseudo-destination D'. Consider a circle of radius $\sqrt{\frac{100 \log n}{\pi n}}$, and hence area $\frac{100 \log n}{n}$ centered around this pseudo-destination. Applying Lemma 60 to the set of n nodes, each such circle contains $\Theta(\log n)$ nodes, w.h.p. In a rare scenario, one of these nodes could potentially be the source node for that flow. However, the circle still has more than one node other than the flow's source. Thus, the flow will select some node within this circle as its destination. Hence, a flow will only be assigned a destination within distance $\sqrt{\frac{100 \log n}{\pi n}}$ from its pseudo-destination. Therefore, a node can only be the destination for flows whose pseudo-destination lies within a distance $\sqrt{\frac{100 \log n}{\pi n}}$ from it. Applying Lemma 60 to the set of n pseudo-destinations, each circle of this size contains $O(\log n)$ pseudo-destinations w.h.p. Thus the number of flows for which any node is the destination is $O(\log n)$ w.h.p. $\qquad\square$

Lemma 2. *For large n, at least one node is the destination for $\Omega(\log n)$ flows with a probability at least $\frac{1}{e}(1 - \frac{1}{e})(1 - \delta)$, where $\delta > 0$ is an arbitrarily small constant.*

Proof. The necessary condition for connectivity in [42] (Theorem 2.1 of [42]) was established by proving that if we consider $R(n)$ such that $\pi R^2(n) = \frac{\log n + b(n)}{n}$, where $\limsup b(n) = b < \infty$, then with positive probability, there exists at least one node x which is isolated, i.e., there is no other node within distance $R(n)$ of x. In the context of [42], this was utilized by interpreting $R(n)$ as transmission range, and thus obtaining a lower bound for connectivity. However, we now exploit that result in a different manner to prove our lemma as follows: Choose $R(n) = \sqrt{\frac{\log n + 1}{\pi n}}$, i.e., $b(n) = b = 1$. Note that in this proof, $R(n)$ is *not* the transmission range; it is merely a chosen distance value. Invoking Theorem 2.1 from [42], with probability p there exists a node x such that there is no other node within a distance $R(n)$ from it, where $\liminf_{n \to \infty} p \geq e^{-b}(1 - e^{-b}) = \frac{1}{e}(1 - \frac{1}{e})$. It follows (see Theorem 2.1 in [42]) that $p \geq (1 - \epsilon)\frac{1}{e}(1 - \frac{1}{e})$, for any $\epsilon > 0$, and sufficiently large n. Call this event $\mathcal{E}_1$.

Conditioned on the occurrence of event $\mathcal{E}_1$, and therefore the existence of such a node x, let us consider the Voronoi tessellation generated by the n nodes. Evidently, the area of

the Voronoi polygon of x is at least $\pi(\frac{R(n)}{2})^2 = \frac{\pi R^2(n)}{4} = \frac{\log n + 1}{4n}$. Note that this tessellation constitutes a spatial partition of the network area. From the definition of the traffic model, it follows that if a flow's pseudo-destination falls within the polygon of node x, then x is selected as that flow's destination, unless x is itself the source of that flow (since a generator (node) is always the nearest node to points within its own Voronoi polygon). Recall that pseudo-destination locations are chosen uniformly at random over the unit torus. Let $X_i, 1 \le i \le n$ be indicator variables such that $X_i = 1$ if x is flow i's destination, and 0 else. Then $\Pr[X_i = 1 | \mathcal{E}_1] = 0$ if x is the source of flow i (and there is exactly one such i).

For all other values of i, x would be selected as flow i's destination if either (1) flow i's pseudo-destination falls in x's Voronoi polygon (the probability of this event is given by the area of x's Voronoi polygon, and is thus at least $\frac{\log n + 1}{4n}$, or (2) if flow i's pseudo-destination falls within the polygon of its own source, and x is the next-nearest node (we can ignore this latter possibility, as we only require a lower bound, and we therefore pretend that x is chosen as the destination of flow i if and only if flow i's pseudo-destination falls within x's Voronoi polygon).

In light of the above, it can be seen that for all i such that x is not the source of flow i: $\Pr[X_i = 1 | \mathcal{E}_1] \ge \frac{\log n + 1}{4n}$. Let $X = \sum\limits_{i:\ x \text{ not source of } i} X_i$. Thus $E[X | \mathcal{E}_1] \ge (1 - \frac{1}{n})\frac{\log n + 1}{4} \ge \frac{\log n}{4}$ for large n. Furthermore, the X_i's are independent. Therefore, application of the Chernoff bound from Lemma 53 (with $\beta = \frac{1}{2}$) yields that:

$$\Pr[X \le \frac{\log n}{8} | \mathcal{E}_1] \le \Pr[X \le \frac{E[X | \mathcal{E}_1]}{2} | \mathcal{E}_1] \le \exp(-\frac{E[X | \mathcal{E}_1]}{8}) \le \exp(-\frac{\log n}{32}) = \frac{1}{n^{\frac{1}{32}}} \quad (2.3)$$

Denote by $\mathcal{E}_2$ the event that some node indeed is destination for at least $\frac{\log n}{8}$ flows. Using (2.3), we obtain that $\Pr[\mathcal{E}_2 | \mathcal{E}_1] \ge 1 - \frac{1}{n^{\frac{1}{32}}}$. Also, $\Pr[\mathcal{E}_2] \ge \Pr[\mathcal{E}_1] \Pr[\mathcal{E}_2 | \mathcal{E}_1]$. Hence at least one node is a destination for $\Omega(\log n)$ flows with a probability at least $(1-\epsilon)e^{-b}(1-e^{-b})(1-\frac{1}{n^{\frac{1}{32}}}) \ge \frac{1}{e}(1 - \frac{1}{e})(1 - \delta)$ for any chosen $\delta > \epsilon$, and sufficiently large n. $\qquad \square$

2.7 A Remark on the Proof Technique

We make a remark on the proof techniques that are used in Chapter 3 and Chapter 4. It is to be noted that many of the intermediate lemmas in the proofs are conditioned on certain desirable events proved to occur w.h.p. in prior lemmas. Let a generic undesirable event

be denoted by $\mathcal{E}_i$ (i.e., $\neg\mathcal{E}_i$ is the desirable event). Note that the following is always true:

$$\Pr[\mathcal{E}_1 \cup \mathcal{E}_2] = \Pr[\mathcal{E}_1] + \Pr[\neg\mathcal{E}_1]\Pr[\mathcal{E}_2|\neg\mathcal{E}_1] \leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2|\neg\mathcal{E}_1] \qquad (2.4)$$

In light of this, it is not hard to see that the probability that even one of the undesirable events from any of these lemmas occurs, can be upper-bounded via by summing up the individual (in some cases, conditional) probability of occurrence of each undesirable event, as bounded by each lemma (i.e., by essentially applying a union bound on the possibly conditional probabilities). Since a proof comprises a small constant number of lemmas, and each lemma proves that the (possibly conditional on previous lemmas) probability of occurrence of some undesirable event goes to 0 (or equivalently shows that the probability of occurrence of the complementary desirable event goes to 1), this sum will also go to zero. Hence, the probability that even one of the undesirable events happens goes to 0. Where not explicitly stated, this union-bound argument is implicitly applied.

Chapter 3

Adjacent (c, f) Assignment

In this chapter, we present capacity results for the adjacent (c, f) assignment model that was introduced in Chapter 2. We begin by defining the adjacent (c, f) assignment model in Section 3.1, and summarize the chapter results in Section 3.2. We present necessary and sufficient conditions for connectivity in Section 3.3. Section 3.4 presents an upper bound on capacity. A capacity-achieving lower bound construction in described in Section 3.5. In Section 3.6, we show how our results for adjacent (c, f) assignment can be used to obtain results for the case of untuned radios. We conclude with a brief discussion on the implications of the capacity result in Section 3.7.

3.1 Model Definition

In the adjacent (c, f) model, the frequency band is divided into c channels numbered 1, 2, ..., c in order of increasing frequency, but an individual interface can only use f channels $(2 \leq f \leq c)$. Prior to deployment, each interface is assigned a block location i uniformly at random from $1, ..., c - f + 1$ and thereafter it can switch between the set $i, ..., i + f - 1$. Thus, the probability that an interface is assigned block location i (where $1 \leq i \leq c - f + 1$) is $\frac{1}{c-f+1}$.

Since channel i occurs in $\min\{i, c - i + 1, f, c - f + 1\}$ blocks, and each block has a probability $\frac{1}{c-f+1}$ of being assigned:

$$\Pr[\text{ a given interface can switch on channel } i] = p_s^{adj}(i) = \frac{\min\{i, c - i + 1, f, c - f + 1\}}{c - f + 1}$$

$$(3.1)$$

Since we consider only single-interface nodes for the results in this chapter, there is a one-to-one mapping between interfaces and nodes. Thus, we often use the term node instead of interface in the following discussion.

18

The probability that a node with block location i can operate on a common channel (we often refer to this as *sharing a channel*) with another randomly chosen node is given by:

$$p_{adj}(i) = \frac{(1 + \min\{i - 1, f - 1\} + \min\{c - f + 1 - i, f - 1\})}{c - f + 1} \qquad (3.2)$$

It is evident that:

$$\min\{\frac{f}{c - f + 1}, 1\} \leq p_{adj}(i) \leq \min\{\frac{2f - 1}{c - f + 1}, 1\} \qquad (3.3)$$

3.2 Summary of Results

We prove the following results:

1. We show that in the regime $c = O(\log n)$, the critical transmission range for connectivity with adjacent (c, f) assignment is $\Theta(\sqrt{\frac{c \log n}{fn}})$.

2. We establish the per-flow capacity under adjacent (c, f) assignment for the regime $c = O(\log n)$ as $\Theta(W\sqrt{\frac{f}{cn \log n}})$.

A preliminary version of the chapter results was reported in [7].

3.3 Conditions for Connectivity

3.3.1 Necessary Condition for Connectivity

We obtain a necessary condition for connectivity through an adaptation of the proof techniques used to obtain the necessary condition for connectivity in [42].

Theorem 1. *With an adjacent (c, f) channel assignment (when $c = O(\log n)$), if $p = \min\{\frac{2f-1}{c-f+1}, 1\}$ and $\pi r^2(n) = \frac{(\log n + b(n))}{pn}$, where $b = \lim\limits_{n \to \infty} b(n) < +\infty$ then:*

$$\liminf_{n \to \infty} \Pr[\ disconnection\ ] \geq e^{-b}(1 - e^{-b})$$

where by disconnection *we imply the event that there is a partition of the network.*

Proof. We present a proof-sketch here. The detailed proof is described in Appendix B.

Given that a node has block location i, the probability that it can operate on a common channel with another random node within its range is given in (3.3), and denoted by $p_{adj}(i)$.

Note that $p_{adj}(i)$ is different for different block locations i primarily because nodes with blocks at the fringes of the band are less likely to share channels with other nodes. Since we are deriving a *necessary* condition for connectivity, it is valid to make the following assumption for the purpose of this proof:

Channel pairs $(i, c - f + i + 1), 1 \leq i \leq f - 1$ possess magical capabilities, such that communication on channel i ends up being visible on channel $c - f + i + 1$, and vice-versa. Thus, if a node has channel i, then it can also communicate with a node that does not share any channel with it, but has channel $c - f + i + 1$. Another way to view this situation is that although nodes are assigned channels as per the adjacent (c, f) model, $c - f + i + 1, 1 \leq i \leq f - 1$ is actually an alias for i. Thus, at the time of network operation, a node having channel $c - f + i + 1, 1 \leq i \leq f - 1$ uses channel i instead (i.e., $c - f + i + 1$ serves as an alias for i).

Under this assumption, $\forall \, i : p_{adj}(i) = \min\{\frac{2f-1}{c-f+1}, 1\}$. If the network is disconnected under this assumption, then it must necessarily be so otherwise. This can be argued as follows: suppose we are given a network instance with nodes assigned adjacent channels as per the adjacent (c, f) model, and we then impose the assumption stated above. Suppose this new network is disconnected. Now the imposed assumption is removed, but the channel block assigned to each node remains unchanged. Then, in the new scenario, some nodes that were earlier able to communicate, will not be able to do so anymore; however those nodes that were incapable of communicating will preserve their status quo. Hence, a necessary condition for the hypothetical network would remain valid even in the actual network.

Therefore, to establish a necessary condition for connectivity with adjacent (c, f) assignment, we estabslish a necessary condition for connectivity in a scenario where we have the additional assumption described above. This proof is an adaptation of a similar proof in [42] (Theorem 2.1 in [42]).

We focus on the disconnection event where singleton sets are partitioned from the rest of network. Recall that $p = \min\{\frac{2f-1}{c-f+1}, 1\}$. When $f \geq \frac{c+2}{3}$, $p = 1$, i.e., any pair of nodes that are within range can communicate with each other as they can operate on at least one common channel, and the necessary condition result from [42] applies directly. Thus, we

need to consider only $f < \frac{c+2}{3}$ for which $p = \frac{2f-1}{c-f+1}$.

The probability that a node x is *isolated*, i.e., cannot communicate with any node is give by $p_1 = (1 - p\pi r^2(n))^{(n-1)}$. We can also obtain an upper bound p_2 on the probability that two nodes x and y are both isolated.

When $\lim_{n\to\infty} \sup \, b(n) < +\infty$, it can be shown that:

$$\Pr[\text{ disconnection }] \geq \sum_x \Pr[x \text{ is only isolated node}]$$

$$\geq \sum_x \Pr[x \text{ isolated }] - \sum_{x,y} \Pr[x \text{ and } y \text{ both isolated }]$$

$$\geq \theta e^{-(b+\epsilon)} - (1+\epsilon)e^{-2(b+\epsilon)}$$

$$\text{for any } \theta < 1, \epsilon > 0, \text{ and large } n$$

$$(3.4)$$

Therefore, if $\lim_{n\to\infty} \sup \, b(n) < +\infty$, the network is asymptotically disconnected with some positive probability. The detailed proof is described in Appendix B. □

Corollary 1. *With an adjacent (c, f) assignment, the critical transmission range for connectivity in the regime $c = O(\log n)$ is $\Omega(\sqrt{\frac{c \log n}{fn}})$.*

Proof. Whenever $f \geq \frac{c+2}{3}$, $p = 1 < \frac{3f}{c}$ in Theorem 1, and the necessary condition require $\pi r^2(n) > \frac{\log n}{n} > \frac{c \log n}{3fn}$. Whenever, $f < \frac{c+2}{3}$, $p = \frac{2f-1}{c-f+1} \leq \frac{3f}{c}$, and the necessary condition again requires that $\pi r^2(n) > \frac{c \log n}{3fn}$. Hence with adjacent (c, f) assignment, connectivity requires that $r(n) = \Omega(\sqrt{\frac{c \log n}{fn}})$. □

3.3.2 Sufficient Condition for Connectivity

It can be shown that having $r(n) = a_1\sqrt{\frac{c \log n}{fn}}$, for some suitable constant a_1, suffices to ensure that the network is asymptotically connected w.h.p. This will be evident from our lower bound construction for capacity. Therefore, the proof is not presented separately.

3.4 Upper Bound on Capacity

We proved in Theorem 1 that to avoid isolated nodes $r(n)$ must be $\Omega(\sqrt{\frac{c \log n}{fn}})$. Then by the connectivity constraint mentioned in Section 2.5 of Chapter 2, the per flow throughput is limited to $O(W\sqrt{\frac{f}{cn \log n}})$.

3.5 Lower Bound on Capacity

We present a constructive proof that achieves a per-flow throughput of $\Omega(W\sqrt{\frac{f}{cn\log n}})$. This construction has similarity to the constructions in [43, 36, 65], but has certain distinctive features that stem from the need to address the channel switching constraints.

The surface of the unit torus is divided into square cells of area $a(n)$ each. The transmission range $r(n)$ is set to $\sqrt{8a(n)}$, thereby ensuring that any node in a given cell is within range of any other node in any adjoining cell. Since we utilize the *Protocol Model* [43], a node C can potentially interfere with an ongoing transmission from node A to node B, only if $BC \leq (1+\Delta)r(n)$. Thus, a transmission by A in a given cell can only be affected by transmissions in cells with some point within a distance $(2+\Delta)r(n)$ from it, and all such cells must lie within a circle of radius $O((1+\Delta)r(n))$. Since Δ is independent of n, the number of cells that interfere with a given cell is only some constant (say γ).

We choose $a(n) = \frac{100c\log n}{fn}$ (and hence $r(n) = \sqrt{\frac{800c\log n}{fn}}$).

The following result follows from an application of Lemma 59:

Lemma 3. *The number of nodes in any cell lies between $\frac{50c\log n}{f}$ and $\frac{150c\log n}{f}$ with probability at least $1 - \frac{50\log n}{n}$.*

Definition 1. (Preferred Channels) *Channels i for which $p_s^{adj}(i) \geq \frac{f}{2c}$ are deemed* preferred *channels.*

For any set of f contiguous channels, at least $\lceil \frac{f}{2} \rceil$ of the channels have $p_s^{adj}(i) \geq \frac{f}{2c}$. Hence, each node can switch on $x \geq \lceil \frac{f}{2} \rceil \geq \frac{f}{2}$ *preferred* channels. Also note that non-preferred channels only occur at the fringes of the frequency band.

Lemma 4. *If there are at least $\frac{50c\log n}{f}$ nodes in every cell $\mathcal{H}$, then at least $12\log n$ nodes in each cell are capable of switching on each of the* preferred *channels, with probability at least $1 - q_1$, where $q_1 = O(\frac{1}{n^2})$.*

Proof. Let us consider one particular cell $\mathcal{H}$, with $x_{\mathcal{H}} \geq \frac{50c\log n}{f}$ nodes. Let $X_{ij} = 1$ if node j can switch on preferred channel i, and 0 else. $\Pr[X_{ij} = 1] = p_s^{adj}(i) \geq \frac{f}{2c}$, since i is a *preferred* channel. For a given i, all the X_{ij}'s are independent. Let $X_i = \sum_{i \in \mathcal{H}} X_{ij}$. Then:

$$E[X_i] = p_s^{adj}(i)x_{\mathcal{H}} \geq 25\log n$$

Applying the Chernoff bound in Lemma 53 (setting $\beta = \frac{1}{2}$) , we obtain:

$$\Pr[X_i \le 12 \log n] \le \Pr[X_i \le \frac{25}{2} \log n] \le \exp(-\frac{25 \log n}{8}) \le \frac{1}{n^{\frac{25}{8}}} \qquad (3.5)$$

The number of preferred channels is at most $c = O(\log n)$. Application of the union bound over all such channels yields:

$$\Pr[X_j \le 25 \log n \text{ for any preferred } j] \le \frac{c}{n^{\frac{25}{8}}} = O(\frac{\log n}{n^{\frac{25}{8}}})$$

Since there are $\frac{1}{a(n)} = \frac{fn}{100c \log n} \le n$ cells, another application of the union bound yields:

$$\Pr[X_i < 12 \log n \text{ in any cell }] = O(\frac{1}{n^2}) \qquad (3.6)$$

$\square$

Each cell indeed has at least $\frac{50c \log n}{f}$ nodes w.h.p. (Lemma 3). Thus, a union bound argument (as was explained in Section 2.7) can be invoked to show that each cell has at least $12 \log n$ nodes on every *preferred* channel w.h.p.

Lemma 5. *If there are at least $\frac{50c \log n}{f}$ nodes in every cell $\mathcal{H}$, then, for all adjacent* preferred *channels i and $i+1$, there are at least $12 \log n$ nodes in each cell capable of switching on both channels i and $i+1$, with probability at least $1 - q_2$, where $q_2 = O(\frac{1}{n^2})$.*

Proof. Let us consider one particular cell $\mathcal{H}$ with $x_{\mathcal{H}}$ nodes, where $x_{\mathcal{H}} \ge \frac{50c \log n}{f}$. Let $X_{ij} = 1$ if node j can switch on both channel i and $i+1$ (where both i and $i+1$ are *preferred*), and 0 else. For a given i, all the X_{ij}'s are independent.

Then $\Pr[X_{ij} = 1] \ge \frac{\lceil \frac{f}{2} \rceil}{c - f + 1} \ge \frac{f}{2c}$. Let $X_i = \sum_{j \in \mathcal{H}} X_{ij}$. Then $E[X_i] \ge 25 \log n$. By application of the Chernoff bound from Lemma 53 (with $\beta = \frac{1}{2}$) , we obtain:

$$\Pr[X_i \le 12 \log n] \le \Pr[X_i \le \frac{25}{2} \log n] \le \exp(-\frac{25 \log n}{8}) \le \frac{1}{n^{\frac{25}{8}}} \qquad (3.7)$$

i cannot take more than $c - 1$ distinct values, and $c - 1 = O(\log n)$. By taking a union bound over all such possibilities, we obtain that $\Pr[X_i \le 12 \log n \text{ for any preferred } i, i+1] \le \frac{(c-1)}{n^{\frac{25}{8}}} = O(\frac{\log n}{n^{\frac{25}{8}}})$. Since there are $\frac{1}{a(n)} = \frac{fn}{100c \log n} < n$ cells, another application of the union

bound yields:

$$\Pr[X_i \leq 12 \log n \text{ in any cell}] = O(\frac{1}{n^2}) \qquad (3.8)$$

$\square$

From Lemma 3, each cell has at least $\frac{50c \log n}{f}$ nodes w.h.p. Thus, each cell has at least $12 \log n$ nodes on every pair of adjacent *preferred* channels $(i, i+1)$ w.h.p.

Lemma 6. *If there are at least $\frac{50c \log n}{f}$ nodes in every cell, and if i and $i + x$ are both preferred channels, where $x \leq \lfloor \frac{f}{2} \rfloor$, then there are at least $12 \log n$ nodes in the cell capable of switching on both channels i and $i + x$ with probability at least $1 - q_3$, where $q_3 = O(\frac{1}{n^2})$.*

Proof. Note that since i is preferred, it follows that $i \geq \lceil \frac{f}{2} \rceil$. A node can switch on both i and $i + x$ if its block location lies between $\max\{1, i + x - f + 1\}$ and i. This probability is $\frac{\min\{i, f - x\}}{c - f + 1}$. Since $x \leq \lfloor \frac{f}{2} \rfloor$, this probability is at least $\frac{\lceil \frac{f}{2} \rceil}{c - f + 1} \geq \frac{f}{2c}$. Thereafter the proof argument is the same as that of Lemma 5. $\square$

3.5.1 Routing

We denote the source of a flow by S, the pseudo-destination by D', and the actual destination by D. We begin by briefly summarizing the routing strategy used in [43]. In [43], one node in each cell was designated the relay for all routes traversing that cell but not originating/terminating in it; a flow's route traversed the cells intersected by the straight line SD' (i.e., they were relayed through the assigned relay nodes in the sequence of cells intersected by the straight-line SD') and thereafter needed to take at most one extra-hop to reach the actual destination D, which necessarily lay either in the same cell as D' or in one of the 8 adjacent cells.

Lemma 7. *The number of straight-line $SD'D$ routes that traverse any cell is $O(n\sqrt{a(n)})$.*

Proof. From Lemma 61 (Appendix E) we know that the number of SD' straight-lines traversing a single cell are $O(n\sqrt{a(n)})$. We must now consider the number of routes whose last $D'D$ hop may enter this cell. If D is in the same cell as D', there is no extra hop. Otherwise, the number of flows for which D' lies in one of the 8 adjacent cells is $O(na(n))$ w.h.p. (since applying Lemma 59 to the set of n pseudo-destinations) yields that the number

of pseudo-destinations in any cell is $O(na(n))$ w.h.p.). Since $na(n) = O(n\sqrt{a(n)})$, the total number of traversing routes is $O(n\sqrt{a(n)})$. □

Hereafter, we shall refer to this routing strategy as straight-line routing, since it basically comprises a straight-line except for the last hop.

If there were no constraints on channel switching, one could envision determining the cells that a route should traverse using a routing strategy similar to that in [43]. We do remark that, even in the absence of switching constraints, in a multi-channel network with c channels where each node has fewer than c interfaces, it does not suffice to designate a single relay node in each cell, as multiple nodes must be concurrently active within a cell to harness the available bandwidth (see [65]).

In the presence of the switching constraints imposed by the adjacent (c, f) assignment, a feasible route must comprise more than just a sequence of nodes from source to destination such that consecutive nodes are with range of each other. Rather, a feasible route must comprise a sequence of nodes $v_0 = S, v_1, ..., v_k, v_{k+1} = D$ such that for all $0 \le i \le k$: (1) v_i and v_{i+1} are with range of each other (2) they can operate on some common channel.

To be able to find such a feasible route w.h.p., the route of a flow may need to traverse a certain minimum number of intermediate nodes (i.e., a feasible sequence of nodes leading from $v_0 = S$ to $v_{k+1} = D$ must have a certain minimum number of nodes). We elaborate further:

We begin by observing that the source must transmit on one of the channels that its interface can switch on. Similarly, the destination must receive on one of the channels that its interface can switch on. Suppose the source uses channel l to transmit, and the destination chooses to use channel r to receive:

We assume w.l.o.g. that $l \le r$. Suppose $r - l = k'\lfloor \frac{f}{2} \rfloor + m$ $(0 \le m < \lfloor \frac{f}{2} \rfloor)$. Thus $k' = \frac{r-l-m}{\lfloor \frac{f}{2} \rfloor} \le \frac{c-1}{\frac{f-1}{2}} = \frac{2(c-1)}{f-1} \le \frac{4c}{f}$. From the model, and the definition of a preferred channel, it follows that, given two preferred channels l and r all channels $l \le i \le r$ must also necessarily be preferred. In light of this, using the result proved in Lemma 6, one can see that it is always possible to transition from l to r in at most $k' + 1 \le \frac{4c}{f} + 1$ steps: $l \to l + \lfloor \frac{f}{2} \rfloor, l + \lfloor \frac{f}{2} \rfloor \to l + 2\lfloor \frac{f}{2} \rfloor, ..., l + k'\lfloor \frac{f}{2} \rfloor \to l + k'\lfloor \frac{f}{2} \rfloor + m = r$.

More specifically, we can find a sequence of nodes $v_0 = S, v_1, v_2, ...v_{k'}, v_{k'+1} = D$ such that v_0 and v_1 both can operate on channel l, v_1 and v_2 can both operate on channel $l + \lfloor \frac{f}{2} \rfloor$

and so on.[1] It is also evident that a sequence of nodes that allows a transition from channel l to channel r must comprise at least $\frac{|r-l|}{f}$ nodes.

More generally, we can try to find a feasible route which comprises a sequence of nodes $v_0 = S, v_1, ...\mathbf{v_i}, ..., \mathbf{v_{i+k'+1}}...v_k = D$, such that (1) $v_0, ..., v_i$ can all operate on channel l, (2) for all $i \leq m < i + k'$: v_m and v_{m+1} can both operate on channel $l + m\lfloor\frac{f}{2}\rfloor$, (3) $v_{i+k'}$ and $v_{i+k'+1}$ can both operate on channel r, and (4) $v_{i+k'+2}, ..., v_k$ can all operate on r. The subsequence $v_i, ..., v_{i+k'+1}$ comprises the transition sequence in this route. Links on this route that lie before the transition sequence use the source channel l to transmit the flow's packets, while links that lie after the transition sequence use the destination channel r. Links $(v_{i+x-1}, v_{i+x}), 1 \leq x \leq k'$ in the transition sequence use channel $l + x\lfloor\frac{f}{2}\rfloor$ for $x \leq k'$ and link $v_{i+k'}, v_{i+k'+1}$ uses channel r.

From the above it is evident that a feasible route must comprise a certain minimum number of intermediate relay nodes, i.e., must traverse a certain mimumum number of hops.

We now address the issue of how the channels l and r are chosen by S and D respectively.

Channel Selection and Transition Initially, after each source has chosen a random destination, the each flow is assigned an initial source channel, as well as a target destination channel in the following manner:

The source S of a flow has an assigned contiguous channel-set (say $(i, ..., i + f - 1)$), while the destination D also has an assigned contiguous channel-set (say $(j, ..., j + f - 1)$). One of the $x \geq \frac{f}{c}$ *preferred* channels available at the source is selected uniformly at random as the source channel. One of the $y \geq \frac{f}{2}$ *preferred* channels available at the destination is selected as the channel on which the flow reaches the destination. The choice of destination channel can be made using any arbitrary criterion from amongst all preferred channels that the destination can operate on.

To ensure that each route has enough hops to assure a feasible transition sequence, we stipulate that the straight-line cell-to-cell path be followed if either the chosen source and destination channels are the same, or if the straight-line segment SD' comprises $h \geq \frac{4c}{f}$ *intermediate* hops. If S and D' (hence also D) lie close to each other, the hop-length of the straight line cell-to-cell path can be much smaller. In this case, a longer *detour* path

---

[1]When $l \geq r$, the transitions are of the form $l \rightarrow l - \lfloor\frac{f}{2}\rfloor, ..., r$.
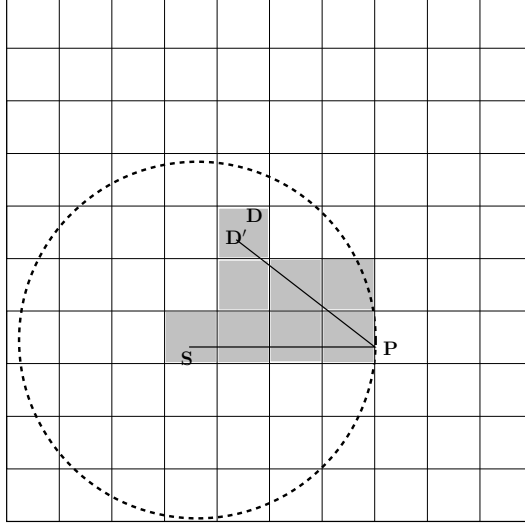
Figure 3.1: Illustration of detour routing

is chosen. Consider a circle of radius $\frac{4c}{f}r(n)$ centered at S. Choose a point on this circle, say P. In the considered $c = O(\log n)$ regime, P can be any point on the circle. The route is obtained by traversing cells along SP and then $PD'D$. This ensures that the route has at least the minimum required hop-length (provided by segment SP). This situation is illustrated in Fig. 3.1. Flows that follow such a detour route shall hereafter be referred to as *detour-routed* flows, whereas the remaining flows (which follow a straight-line route) will be referred to as *non-detour-routed* flows.

The route of a flow comprises two phases: a *progress-on-source-channel* phase, and a *transition* phase. Intuitively, while in the *progress-on-source-channel* phase, the flow's packets are transmitted at each hop on the chosen source-channel l. Once in the *transition* phase, the packets get transmitted along a sequence of channels that constitute a transition from l to r, as was described earlier. Once the transition sequence has reached channel r, the packets are transmitted along any remaining hops on r, till they are received at the destination.

The initial hops of the route of a non-detour-routed flow constitute the *progress-on-source-channel* phase. The flow remains in this phase till there are only $\lceil \frac{4c}{f} \rceil$ intermediate hops left to the destination. At this point, it enters *transition* mode. A detour-routed flow is always in *transition* mode.

Lemma 8. *Suppose the event addressed in Lemma 6 holds. Suppose a flow's selected*

preferred *source channel is l and its selected* preferred *destination channel is r. Then, after having traversed* $h \geq \lceil \frac{4c}{f} \rceil + 1$ *cells (recall that* $2 \leq f \leq c$*) , it is guaranteed to have made the transition.*

Proof. The event considered in Lemma 6 is that each cell has at least $12 \log n$ nodes on each pair of preferred channels (i, x), for all $x \leq \lfloor \frac{f}{2} \rfloor$. Given that the chosen source channel is l, the flow packets are transmitted on l on those hops where the flow is in *progress-on-source-channel* mode. When the flow moves into *transition* mode, the first relay node in this phase chooses as next-hop a node having channel pair $(l, l + \lfloor \frac{f}{2} \rfloor)$ in the next cell (the exact method for choosing relay nodes is described later), and transmits the flow's packets to it using channel l. This node then chooses a next hop having channel pair $(l + \lfloor \frac{f}{2} \rfloor, l + 2 \lfloor \frac{f}{2} \rfloor)$, and sends packets to it over channel $l + \lfloor \frac{f}{2} \rfloor$, and the process continues till the flow has found a transition into the chosen destination channel r. This requires at most $\lceil \frac{4c}{f} \rceil$ intermediate hops, which are obtained by traversing at most $\lceil \frac{4c}{f} \rceil + 1$ cells. Once the transition to destination channel r is done, flow packets are transmitted on channel r for the remaining hops (if any) to the destination. $\qquad \square$

The event considered in Lemma 6 holds w.h.p., and therefore, each flow will be able to find such a transition sequence w.h.p.

Lemma 9. *If the number of distinct flows traversing any cell is x in case of pure straight-line routing, it is at most* $x + O(n \frac{c^2}{f^2} r^2(n)) \implies x + O(\log^4 n)$ *even with detour routing* [2].

Proof. Since a detour route lies within a circle of radius $\frac{4c}{f} r(n)$ around the source, the extra detour-routed flows that may possibly pass through a cell (compared to the case where only straight-line routing is performed) are those whose sources lie within a distance $\frac{4c}{f} r(n)$ from this cell. All such possible sources fall within a circle of radius $(\frac{4c}{f} + 1) r(n)$, and hence area $a_c(n) = \pi (\frac{4c}{f} + 1)^2 r(n)^2$. Any circle of this radius has $O(n a_c(n))$ nodes, and hence at most $O(n a_c(n))$ sources w.h.p. (Lemma 60). Therefore, the number of detour-routed flows that traverse the cell is $O(n a_c(n)) = O(n \frac{c^2}{f^2} r^2(n))$, and the total number of flows is $x + O(n \frac{c^2}{f^2} r^2(n)) \implies x + O(\log^4 n)$ w.h.p. $\qquad \square$

---

[2]This is a loose upper bound. The actual number of detour-routed flows traversing a cell is much smaller.

Lemma 10. *The number of flow-links traversing any cell in* transition *phase (counting repeat traversals separately) is* $O(\log^4 n)$ *w.h.p.*

Proof. First let us account for the SD' stretch of each flow's route, without considering the possible additional last hop. We account for it explicitly later in this proof.

By our construction, a non-detour routed flow enters the *transition* phase only when it is $\lceil \frac{4c}{f} \rceil$ intermediate hops away from its destination. All such flows must have their pseudo-destinations within a circle of radius $\Theta(\frac{c}{f} r(n))$ centered in the cell. The number of destinations that lie within a circle of radius $\Theta(\frac{c}{f}) r(n)$ from the cell is $\Theta(n(\frac{c}{f})^2 r^2(n)) \implies O(\frac{c^3}{f^3} \log n)$ w.h.p., (by suitable choice of $\alpha(n) = O(\frac{c^3}{f^3})$ in Lemma 60). Thus the number of non-detour routed flows that may traverse a cell is $O(\frac{c^3}{f^3} \log n)$.

A detour-routed flow is always in *transition* phase. By Lemma 9, there are $O(\log^4 n)$ such flows traversing any cell. Each such flow can only traverse a cell at most twice along the SPD' stretch. This yields $O(\log^4 n)$ detour-routed flows (including repeat traversals).

Also, the cell may be traversed/re-traversed by some flows on their additional last hop. There are $O(na(n))$ pseudo-destinations in the adjacent cells w.h.p., and thus $O(na(n)) = O(\frac{c \log n}{f}) \implies O(\log^2 n)$ such last hop flow traversals. Thus the number of flows transitioning in any cell is $O(\frac{c^3}{f^3} \log n)) + O(\log^4 n) + O(\log^2 n)$. Taking note that $c = O(\log n)$, it follows that the number of flows traversing the cell while in their *transition* phase is $O(\log^4 n)$ w.h.p. $\qquad\square$

Relay Node Selection We now describe how a relay node is assigned to a flow's route in each cell.

A flow-link is said to *enter* a cell $\mathcal{H}$ on a channel j if the flow's route includes a hop (link) (v_{i-1}, v_i), where v_{i-1} is in a cell adjacent to $\mathcal{H}$, v_i is in $\mathcal{H}$, and v_{i-1} transmits the flow's packets to v_i using channel j (this naturally implies that both v_{i-1} and v_i can operate on channel j). Similarly, a flow-link is said to leave a cell $\mathcal{H}$ on channel j if the route includes a link (v_i, v_{i+1}), where v_i is in $\mathcal{H}$, v_{i+1} is in a cell adjacent to $\mathcal{H}$, and v_i transmits the flow's packets to v_{i+1} using channel i

When a flow-link must enter a cell in *progress-on-source-channel* phase on a certain channel, then, amongst all nodes in that cell capable of switching on that channel, it is assigned to the node which has the least number of flow-links entering on that channel assigned to it so far. In the *transition* phase of a flow, a flow-link may need to be assigned

a relay node that can operate on a specific pair of channels (to facilitate transition). It can be assigned to any node in the cell that satisfies the requirement. Similarly, once a flow in *transition* phase has already completed to the transition, the remaining links on the route will enter the remaining cells on its route on the destination channel. Such a flow-link can be assigned to any node in the cell that can switch on the destination channel.

3.5.2 Load Balance within a Cell

Recall that each cell has $O(na(n))$ nodes w.h.p., and $O(n\sqrt{a(n)})$ flows traversing it w.h.p.

Per-Channel Load

Lemma 11. *The number of flow-links that enter any cell on any single channel is* $O(\frac{n\sqrt{a(n)}}{c})$ *w.h.p.*

Proof. Consider a cell $\mathcal{H}$.

A flow's route may enter a channel i in the cell in any of the following circumstances:

1. The flow's source channel is i and it is in the *progress-on-source-channel* phase

2. The flow's route is in the *transition* phase, and transitioning through i

3. The flow's route is in the *transition* phase, its destination channel is i, and it has already made a transition

We first account for the flow-routes in *progress-on-source-channel* phase:

From our construction, and by our choice of $a(n)$, each flow stays in *progress-on-source-channel* phase, till there are $\lceil \frac{4c}{f} \rceil$ intermediate hops left to the destination. Thus, a flow is on its source channel in a given cell if its destination is more than $\lceil \frac{4c}{f} \rceil$ intermediate hops away.

Denote the number of flow-routes traversing the cell in *progress-on-source channel* phase by m. Then $m = O(n\sqrt{a(n)})$ (from Lemma 7).

Let X_{ij} be an indicator variable which is 1 if flow-route j enters the cell on channel i, and is 0 else.

From the model definition, each source's interface is assigned a block of f contiguous channels in an i.i.d. manner, and it chooses one channel uniformly from $x \geq \lceil \frac{f}{2} \rceil \geq \frac{f}{2}$

30

preferred channels in this channel block. Furthermore, the sequence of cells traversed by a flow's route is chosen in a manner independent of the channels the source can switch on.

Hence, the probability that a flow-route in *progress-on-source-channel* phase is on a particular *preferred* channel i is at most $\frac{2p_s^{adj}(i)}{f} = \left(\frac{2}{f}\right)\left(\frac{\min\{f,c-f+1,i,c-i+1\}}{c-f+1}\right) \leq \left(\frac{2}{f}\right)\left(\frac{\min\{f,c-f+1\}}{c-f+1}\right) = \frac{2}{\max\{f,c-f+1\}} \leq \frac{4}{c}$, yielding:

$$\frac{1}{c} \leq \Pr[X_{ij} = 1] \leq \frac{4}{c}$$

$X_i = \sum_j X_{ij}$ denotes the number of flow-routes in *progress-on-source-channel* phase that enter the cell on channel i. Evidently:

$$\frac{m}{c} \leq E[X_i] \leq \frac{4m}{c}$$

The X_{ij}'s are i.i.d. random variables for a given i, as each flow's source channel is chosen in an i.i.d. manner (though they may not be independent for different i, since $X_{ij} = 1 \implies X_{ik} = 0\ \forall k \neq i$). Hence we may set $(1+\beta)E[X_i] = \max\{\frac{4e^2m}{c}, 3\log n\}$ (note that $\beta \geq e^2 - 1 > 0$), and apply the Chernoff bound from Lemma 51 to obtain:

$$
\begin{aligned}
\Pr[X_i \geq \max\{\frac{4e^2m}{c}, 3\log n\}] &\leq \left(\frac{e^\beta}{(1+\beta)^{(1+\beta)}}\right)^{E[X_i]} \\
&\leq \left(\frac{e}{(1+\beta)}\right)^{(1+\beta)E[X_i]} \\
&\leq \left(\frac{eE[X_i]}{\max\{\frac{4e^2m}{c}, 3\log n\}}\right)^{(1+\beta)E[X_i]} \\
&\leq \exp(-(1+\beta)E[X_i]) \leq \exp(-\max\{\frac{4e^2m}{c}, 3\log n\})
\end{aligned}
\tag{3.9}
$$

The number of *preferred* channels c_{pref} cannot exceed c. Applying the union bound over the $c_{pref} \leq c$ preferred channels, the probability that there are $\max\{\frac{4e^2m}{c}, 3\log n\}$ or more flow-links entering on any single channel is at most $c\exp(-\max\{\frac{4e^2m}{c}, 3\log n\})$. Taking another union bound over all $\frac{1}{a(n)} = \frac{fn}{100c\log n}$ cells, the probability this happens in any cell of the network is less than $\frac{fn}{100\log n}\exp(-\max\{\frac{4e^2m}{c}, 3\log n\}) = O(\frac{1}{n^2})$.

Observing that $\max\{\frac{4e^2m}{c}, 3\log n\} = O(\frac{n\sqrt{a(n)}}{c})$, this proves that the number of non-transitioning flows that enter any cell on a given channel is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p.

We now need to account for the fact that some of the flow-routes may be in the *transi-*

tion phase, and may either be transitioning through an intermediate channel or may have transitioned to the destination channel. From Lemma 10 the number of flow-links for such flows which traverse the cell (counting repeat traversals separately) is $O(\log^4 n)$ w.h.p. Even if they were all to enter on the same channel, the additional contribution to the load would be $O(\log^4 n)$.

Hence the per-channel load in all cells is at most $O(\frac{n\sqrt{a(n)}}{c}) + O(\log^4 n) \implies O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. □

Lemma 12. *The number of flow-links that leave any given cell on any single channel is* $O(\frac{n\sqrt{a(n)}}{c})$ *w.h.p.*

Proof. The flows whose routes leave a cell fall into two categories: (1) those that originate at some node in the cell, and (2) those that entered the cell but did not terminate there (i.e., were relayed through the cell). The former can be no more than the number of nodes in the cell, i.e. $\Theta(na(n)) = \Theta(\frac{c\log n}{f}) = O(\log^2 n)$. For the latter, note that any flow-link that leaves the cell, must then enter one of the 8 adjacent cells. Thus, the former can be no more than 8 times the maximum number of flow-links entering a cell on any one channel, which has been established as $O(\frac{n\sqrt{a(n)}}{c}) = O(\sqrt{\frac{n\log n}{cf}})$ in Lemma 11. Hence, the total number of flow-links leaving any given cell on a given channel is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. □

Per-Node Load

Lemma 13. *The number of flow-links that are assigned to any single node in any cell is* $O(\frac{n\sqrt{a(n)}}{c})$ *w.h.p.*

Proof. A node is always assigned an outgoing link for the single flow for which it is the source. A node is also assigned an incoming link for each flow for which it is the destination (any such flows terminate in that cell), and there are $O(\log n)$ such flows for any node w.h.p. (from Lemma 1).

Additionally, a node may act as a relay node on the routes of other flows. For each such flow, it is assigned an incoming and an outgoing link (as it must receive the flow's packets, and then transmit them on to a next hop node).

It may be assigned as relay for some flow-routes that are in the *transition* phase, and for which it serves as one of the nodes in the channel-transition sequence, or it may be assigned

32

as relay for some flow-routes in *transition* phase which have completed the transition, if it can operate on their destination channel. From Lemma 10 there are $O(\log^4 n)$ such flow-links traversing a cell w.h.p. (counting possible repeat traversal by some detour-routed flows, as well as any additional last hop traversals separately). Resultantly, the number of such flow-links assigned to a node is $O(\log^4 n)$.

It may also be assigned as relay for flow-routes that are in *progress-on-source-channel* phase while they traverse the cell. We have already established in Lemma 11, that the number of flows that enter on a given channel in any cell is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. From our routing and channel transition strategy, flow-links in the *progress-on-source-channel* phase of a route are always operated on the source's selected *preferred* channel. From Lemma 4, there are at least $12 \log n$ nodes on each *preferred* channel in each cell w.h.p. As per our previously described relay node selection strategy, when a relay node is to be assigned to an incoming flow-link in *progress-on-source-channel* phase in a cell on a certain channel, then amongst all nodes in the cell capable of switching on that channel, it is assigned to the node which has the least number of entering flow-links assigned on that channel so far. By using such an assignment strategy, it follows that no node can have more than $O(\frac{n\sqrt{a(n)}}{c\log n})$ such flow-links assigned on any single channel, and no more than $O(\frac{fn\sqrt{a(n)}}{c\log n}) \implies O(\frac{n\sqrt{a(n)}}{c})$ such flow-links assigned overall (recall that $c = O(\log n)$, and $f \leq c$).

For each incoming flow-link assigned to a node for relaying, there is a corresponding outgoing flow-link (as the node is a relay).

Thus, the resultant number of assigned flows per node is $1 + O(\log n) + O(\log^4 n) + O(\frac{n\sqrt{a(n)}}{c}) = O(\frac{n\sqrt{a(n)}}{c})$. $\qquad\qquad\square$

3.5.3 Transmission Schedule

We noted earlier that each cell can face interference from at most a constant number γ of nearby cells, where γ is a constant. The resultant cell-interference graph has a chromatic number at most $1 + \gamma$. Therefore, it is possible to obtain a global interference-free TDMA schedule having $1 + \gamma$ time slots in each round. In any slot, if a cell is active, then all cells that interfere with it are inactive. The next issue is that of intra-cell scheduling. We need to schedule transmissions so as to ensure that, at any time instant, there is at most one transmission on any given channel in the cell. Besides, we also need to ensure that no node

is expected to transmit or receive more than one packet at any time instant. We use the following procedure to obtain an intra-cell schedule:

We construct a conflict graph based on the nodes in the active cell, and its adjacent cells , as follows:

We create a separate vertex for each flow-link leaving the cell (note that the hop-sender of each such flow-link shall lie in the active cell, and the hop-receiver shall lie in one of the adjacent cells). Since the flow-link operates on an assigned channel, each vertex in the graph has an implicit associated channel. Besides, each vertex has an associated pair of nodes corresponding to the hop endpoints. Two vertices are connected by an edge if either (1) they have the same associated channel, or (2) at least one of their associated nodes is the same.

The scheduling problem reduces to obtaining a vertex-coloring of this graph. If we have a vertex coloring, then it ensures that (1) a node is never simultaneously sending/receiving for more than one flow-link (2) no two flow-links on the same channel are active simultaneously. The number of neighbors of a graph vertex is upper bounded by the number of flow-links leaving the active cell on that channel, and the number of flow-links assigned to the flow's two hop endpoints (both hop-sender and hop-receiver). From Lemma 12 and Lemma 13, the degree of the conflict graph is $O(\frac{n\sqrt{a(n)}}{c}) + O(\frac{n\sqrt{a(n)}}{c}) \implies O(\frac{n\sqrt{a(n)}}{c})$. Since any graph with maximum degree d has chromatic number at most $d+1$, the conflict graph can be colored in $O(\frac{n\sqrt{a(n)}}{c})$ colors.

Therefore, the cell-slot can be divided into $O(\frac{n\sqrt{a(n)}}{c}) = O(\frac{\sqrt{\frac{cn\log n}{f}}}{c})$ equal length sub-slots, and all flow-links get a sub-slot for transmission.

This yields that each flow will get $\Omega(W\sqrt{\frac{f}{cn\log n}})$ throughput.

Combining this with the upper bound from Section 3.4, we obtain the following theorem:

Theorem 2. *With an adjacent (c, f)-channel assignment, where $c = O(\log n)$, the network capacity is $\Theta(W\sqrt{\frac{f}{cn\log n}})$ per flow.*

3.6   The Case of Untuned Radios

It was proposed in [93] that extremely inexpensive wireless devices can be manufactured if it is possible to handle untuned radios whose operating frequency may lie randomly within some band. A random network coding based approach was described in [93] to relay

Center frequency (uniformly distributed over (F_1, F_2))

B
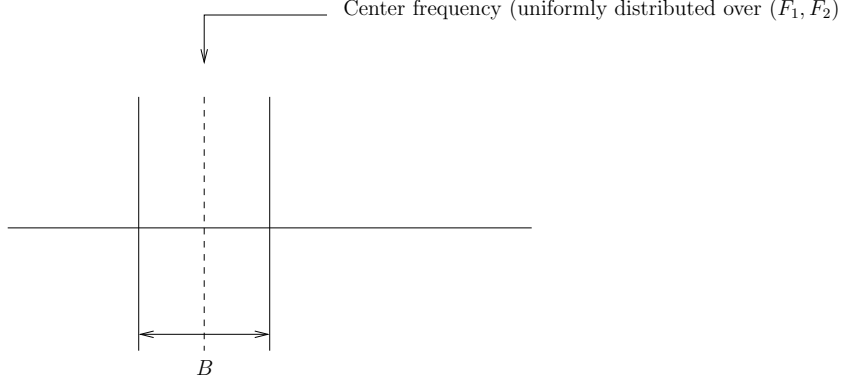
Figure 3.2: The Untuned Radio Model

information between a single source-destination pair using such devices as relays. In this section, we obtain capacity results for a randomly deployed network of n nodes with one untuned radio each, with our assumed model, i.e., n random source-destination pairs, and store-and-forward routing.

The untuned channel model is as follows: each node possesses a transceiver with center frequency uniformly distributed in the range (F_1, F_2), and admits a spectral bandwidth B (Fig. 3.2). Let $c = \lfloor \frac{F_2 - F_1}{B} \rfloor$. Then c is the maximum number of disjoint channels that could be possibly obtained if each channel occupied a frequency band of width B. For simplicity, the rest of the discussion assumes that $c = \lfloor \frac{F_2 - F_1}{B} \rfloor = \frac{F_2 - F_1}{B}$ (i.e., the interval (F_1, F_2) is chosen to be a multiple of B).

However the channels of operation of these radios are untuned and hence partially ovelapping, rather than disjoint. As per the assumption in [93], two nodes can communicate directly if the center frequency of one is admitted by the other, i.e., if there is at least 50% overlap between two channels, communication is possible. We consider the issue of capacity of a network of n nodes, deployed uniformly at random, where each node has an untuned radio, and each node is the source of one flow, with a randomly chosen destination.

Even though each node only possesses a single radio and stays on a single sub-band, due to the partial overlap between sub-bands, it is still possible to ensure that any pair of nodes will be connected via some path. Contrast this to the case of orthogonal channels, where we argued in Section 2.5 that when $f = 1$, and $c > 1$, some pairs of nodes are disconnected from each other because they do not share a channel. It is possible to map the partial overlap feature of the untuned channel case to adjacent $(2c+2, 3)$ and $(4c+1, 2)$
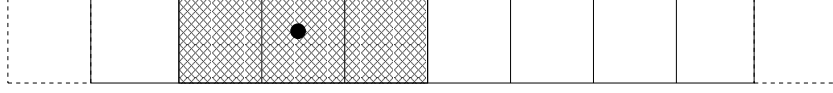
Figure 3.3: Untuned Radios: Upper Bound via virtual $(2c + 2, 3)$ channelization

assignment, and obtain upper and lower bounds. Note that $f \geq 2$ allows for all nodes to be connected, even with orthogonal channels.

3.6.1 Upper Bound on Capacity

We map the untuned radio scenario to a scenario involving $(2c + 2, 3)$ adjacent channel assignment (Fig. 3.3).

We perform a virtual channelization of the band (F_1, F_2) into $2c$ orthogonal sub-bands. We add an additional (virtual) sub-band of the same width at each end of the band, to get $2c + 2$ orthogonal channels, numbered $1, ..., 2c + 2$. Thus 1 and $2c + 2$ are the artificially added channels. If a radio's center frequency lies within virtual channel i, it is associated with virtual channel block $(i - 1, i, i + 1)$, and $i - 1$ is called its primary virtual channel. Thus the primary channel can only be one of $1, 2, ..., 2c$ (since the center frequency can only fall in $2, .., 2c + 1$). If a node's primary channel is i, it is capable of communicating with all nodes with primary virtual channel $i - 2 \leq j \leq i + 2$ in the virtual channelization. In the actual situation, the node with the untuned radio would be able to communicate with some subset of those nodes. Thus, if a pair of nodes cannot communicate directly in the virtual channelization, they cannot do so in the actual situation either, and disconnection events in the former are preserved in the latter. The probability that a node has virtual channel block $(j, j + 1, j + 2)$ is $\frac{1}{2c}$, i.e., the same as for adjacent $(2c + 2, 3)$ assignment, and the assignment of each node is independent. Therefore, the necessary condition for the (virtual) $(2c + 2, 3)$ assignment continues to hold for the corresponding untuned radio case. This yields an upper bound on capacity of $O(W \sqrt{\frac{1}{cn \log n}})$.

3.6.2 Lower Bound on Capacity

It can be shown that a schedule constructed for an adjacent $(4c + 1, 2)$ assignment can be used almost as-is with untuned radios (except that the number of subslots in the cell-slot must increase by a constant factor to avoid interference due to overlap).

36
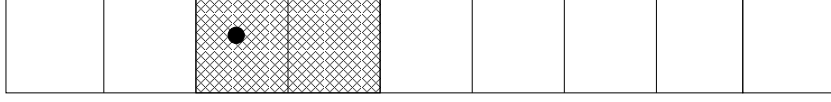
Figure 3.4: Untuned Radios: Lower Bound via virtual $(4c + 1, 2)$ channelization

We perform a virtual channelization of the band (F_1, F_2) into $4c + 1$ orthogonal sub-bands. If a radio's center frequency lies within virtual channel i, it is associated with virtual channel block $(i, i+1)$, and i is called its primary virtual channel.

Thus, if a pair of nodes can operate on a common channel in the virtual channelization, then they are always capable of direct communication in the actual untuned radio situation. The probability that a radio has virtual channel block $(i, i+1)$ is $\frac{1}{4c}$, which is the same as for adjacent $(4c+1, 2)$ assignment, and the assignment of each node is independent. In the adjacent $(4c+1, 2)$ assignment, all channel are orthogonal and can operate concurrently. With untuned radios, we assume that two nodes can interfere if there is some spectral overlap. Thus, a transmission by a node on center frequency F can interfere with transmissions by nodes with center frequency in the range $(F - B, F + B)$. Hence, the transmission schedule for untuned radios is made to follow the additional constraint that if a node with primary virtual channel i is active then no node with primary channel $i - 5 \leq j \leq i + 5$ should be active simultaneously. This can decrease capacity by a factor of 11, but would not affect the order of the asymptotic results. Also, in the actual network involving untuned radios, a transceiver can use upto $B = \frac{F_2 - F_1}{c}$ spectral bandwidth, while in the adjacent $(4c + 1, 2)$ case, it would be $\frac{F_2 - F_1}{4c+1}$, leading to the possibility of having a higher data-rate in the former, given the same transmission power, modulation, etc. However this can only affect capacity by a small constant factor, which does not affect the order of the results.

In the adjacent $(4c + 1, 2)$ case, our construction performs transitions to ensure that a source on channels $(i, i+1)$ and a destination on channels $(i+j, i+j+1)$ can communicate over $j \leq 4c$ hops. In the untuned radio case, transitioning occurs through nodes that provide the required *virtual channel pair*, and the same transition strategy as for $(4c+1, 2)$ assignment continues to work. Hence the capacity is $\Omega(W\sqrt{\frac{1}{cn \log n}})$ per flow.

We re-emphasize that even though $f = 1$, the untuned nature of the radios allows for a progressive shift in the frequency over which the packet gets transmitted, thereby allowing a step-by-step transition from the source's center frequency to a frequency admitted

by the destination. The adjacent (c, f) model captures this progressive frequency-shift characteristic, and is thus able to model the untuned radio situation.

The upper and lower bounds proved in this section lead to the following:

Theorem 3. *In the regime $c = O(\log n)$, the capacity of a randomly deployed network of untuned radios is $\Theta(W\sqrt{\frac{1}{cn\log n}})$ per flow.*

3.7   Discussion

The capacity-achieving construction provides some useful insights. As is intuitive, when all nodes cannot switch on all channels, the transmission range needs to be larger to preserve network connectivity. This leads to a loss of capacity compared to the case of unconstrained switching. Also, it may no longer be possible to use the straight-line path towards the destination, and a flow may need to traverse a larger number of hops (*detour routing*) in order to ensure that the destination is reached. However, when the number of channels is much smaller than the number of nodes, the increase in the length of the routes is not asymptotically significant, and only affects the capacity by a constant factor. Taking all factors into account, given situations where each radio-interface can only be manufactured to switch on f channels out of a total of c available channels (where $c = O(\log n)$), it is beneficial in the asymptotic regime to attempt to use all channels by assigning different channel subsets to different nodes, rather than follow the naive approach of using the same f channels at all nodes. In the latter case, the per-flow capacity would be reduced to $\Theta(W\frac{f}{c\sqrt{n\log n}})$. Thus, the *use-all-channels* approach outperforms the *f-common-channels* approach by a factor of $\Theta(\sqrt{\frac{c}{f}})$. For instance, even when $f = 2$, utilizing all channels yields a capacity of the order of $\sqrt{c}$ channels.

Chapter 4

Random (c, f) Assignment

In this chapter, we present connectivity and capacity results for the random (c, f) assignment model that was introduced in Chapter 2. We begin by defining the random (c, f) assignment model in Section 4.1, and thereafter summarize the chapter results in Section 4.2. In Section 4.3, we state and prove some preliminary results used by subsequent proofs. We present necessary and sufficient conditions for connectivity in Section 4.4. Section 4.5 presents an upper bound on capacity. In Section 4.6 we describe a sub-optimal lower bound construction for capacity. The optimal lower bound construction in described in Section 4.7. Finally, in Section 4.8, we discuss the implications of the capacity result, and the insights that can be obtained from it.

4.1 Model Definition

In the random (c, f) assignment model, each radio-interface is assigned a subset of f channels from a total of c available channels $(2 \leq f \leq c)$ uniformly at random from all such possible subsets. This leads to the following:[1]

$$\Pr[\text{ a given interface can switch on channel } i] = p_s^{rnd}(i) = \frac{f}{c} = p_s^{rnd}, \ \forall i \qquad (4.1)$$

---

[1] The number of ways of selecting k objects from a set of m objects, i.e., $\binom{m}{k}$ is usually defined as $\frac{m!}{k!(m-k)!}$ for $m \geq k \geq 0$. For $k > m \geq 0$ or $k > 0 > m$, one can uniformly define $\binom{m}{k}$ to be 0, as there exists no way of selecting k objects from a set of m objects under these circumstances. In this chapter, we use this convention for notational convenience. It is also to be noted that the expression $\prod_{i=1}^{k} \left(\frac{m-k+i}{i} \right)$ yields $\frac{m!}{k!(m-k)!}$ for $0 < k \leq m$ and is 0 for $k > m \geq 0$.

Pr[two given interfaces can switch on at least one common channel] $= p_{rnd}$

$$= 1 - \frac{\binom{c-f}{f}}{\binom{c}{f}} = 1 - \left(1 - \frac{f}{c}\right)\left(1 - \frac{f}{c-1}\right)\cdots\left(1 - \frac{f}{c-f+1}\right) \qquad (4.2)$$

Evidently: $f \geq c - f + 1 \implies p_{rnd} = 1$.

Since we consider only single-interface nodes for the results in this chapter, there is a one-to-one mapping between interfaces and nodes. Thus, as also in Chapter 3, we often use the term node instead of interface in the following discussion.

4.2 Summary of Results

We prove the following results:

1. We show that in the regime $c = O(\log n)$, the critical range for connectivity with random (c, f) assignment is $\Theta(\sqrt{\frac{\log n}{p_{rnd}n}})$.

2. We establish the per-flow capacity with random (c, f) assignment for the regime $c = O(\log n)$ as $\Theta(W\sqrt{\frac{p_{rnd}}{n \log n}})$.

It can be shown that $p_{rnd} \geq 1 - e^{-\frac{f^2}{c}}$. Hence, the implication of this capacity result is that, when $f = \Omega(\sqrt{c})$, random (c, f) assignment yields capacity of the same order as attainable via unconstrained switching. *Thus, for the random (c, f) assignment model, $\sqrt{c}$-switchability is sufficient to make order-optimal use of all c channels, when $c = O(\log n)$.*

A preliminary version of the chapter results was reported in [7, 6].

4.3 Preliminaries

In this section, we state and prove some results that are required for the proofs that follow.

Lemma 14. *For $c \geq 2$, and $2 \leq f \leq c$:*

$$\frac{cp_{rnd}}{f} \leq \min\{\frac{c}{f}, 2f\} \qquad (4.3)$$

Proof. Since $p_{rnd} \leq 1$, it follows that:

$$\frac{cp_{rnd}}{f} \leq \frac{c}{f} \tag{4.4}$$

If $f \geq \sqrt{\frac{c}{2}}$:

$$\frac{cp_{rnd}}{f} \leq \sqrt{2c} \leq 2f \quad \because p_{rnd} \leq 1 \tag{4.5}$$

Now consider the case $f < \sqrt{\frac{c}{2}}$, i.e., $\frac{2f^2}{c} < 1$. It is to be noted that $f < \sqrt{\frac{c}{2}} \implies \frac{2f}{c} < 1$ for all $c \geq 2$. We take note of the following inequality:

$$
\begin{aligned}
\ln\left((1 - \frac{2f}{c})^f\right) &= f \ln(1 - \frac{2f}{c}) \\
&= f\left[\left(-\frac{2f}{c}\right) - \frac{1}{2}\left(-\frac{2f}{c}\right)^2 + \frac{1}{3}\left(-\frac{2f}{c}\right)^3 - \dots\right] \\
&= \left(-\frac{2f^2}{c}\right) - \frac{1}{2f}\left(-\frac{2f^2}{c}\right)^2 + \frac{1}{3f^2}\left(-\frac{2f^2}{c}\right)^3 - \dots \\
&\geq \left(-\frac{2f^2}{c}\right) - \frac{1}{2}\left(-\frac{2f^2}{c}\right)^2 + \frac{1}{3}\left(-\frac{2f^2}{c}\right)^3 - \dots \\
&= \ln(1 - \frac{2f^2}{c}) \\
\therefore (1 - \frac{2f}{c})^f &\geq 1 - \frac{2f^2}{c} \quad \text{(since } \ln x \text{ is an increasing function of } x)
\end{aligned}
\tag{4.6}
$$

Noting that $c - f + 1 > \frac{c}{2}$ and $c - f + 1 > f$ whenever $f < \sqrt{\frac{c}{2}}$, we obtain that:

$$
\begin{aligned}
1 - p_{rnd} &= \left(1 - \frac{f}{c}\right)\left(1 - \frac{f}{c-1}\right)\dots\left(1 - \frac{f}{c-f+1}\right) \\
&\geq (1 - \frac{f}{c-f+1})^f > (1 - \frac{2f}{c})^f \geq 1 - \frac{2f^2}{c} \quad \text{using (4.6))} \\
\therefore p_{rnd} &\leq \frac{2f^2}{c} \\
\therefore \frac{cp_{rnd}}{f} &\leq 2f
\end{aligned}
\tag{4.7}
$$

From (4.4), (4.5) and (4.7):

$$\frac{cp_{rnd}}{f} \leq \min\{\frac{c}{f}, 2f\}$$

$\square$

Lemma 15. $\min\{\frac{c}{f}, 2f\} \leq \sqrt{2c}$

Proof. For a given c, we have $2 \leq f \leq c$. Thus, given c, $\frac{c}{f}$ is a monotonically decreasing

function of f, while $2f$ is a monotonically increasing function of f. $\frac{c}{f} = 2f = \sqrt{2c}$ at $f = \sqrt{\frac{c}{2}}$. For $f \leq \sqrt{\frac{c}{2}}$, $\min\{\frac{c}{f}, 2f\} = 2f \leq \sqrt{2c}$, and for $f > \sqrt{\frac{c}{2}}$, $\min\{\frac{c}{f}, 2f\} = \frac{c}{f} \leq \sqrt{2c}$. Thus $\min\{\frac{c}{f}, 2f\} \leq \sqrt{2c}$. $\qquad\square$

Lemma 16. *The following inequality holds for all $2 \leq f \leq c$:*

$$\left(\frac{2\binom{c-f}{f} - \binom{c-2f}{f}}{\binom{c}{f}} \right) \leq 1 - \frac{p_{rnd}^2}{40}$$

Proof. We begin by observing that $\frac{\binom{c-f}{f}}{\binom{c}{f}} = 1 - p_{rnd}$.

Consider the following three cases:

Case 1: $f \geq c - f + 1$

This implies that $p_{rnd} = 1$. Noting that the L.H.S cannot exceed $= 2(1 - p_{rnd}) = 0$, the result follows trivially.

Case 2: $\frac{c-f+1}{2} \leq f < c - f + 1$

This implies that $\frac{\binom{c-2f}{f}}{\binom{c}{f}} = 0$. Moreover:

$$\frac{\binom{c-f}{f}}{\binom{c}{f}} = \prod_{i=1}^{f} \left(1 - \frac{f}{c-i+1}\right) \leq \left(1 - \frac{f}{c-f+2}\right)\left(1 - \frac{f}{c-f+1}\right)$$

$$\leq \left(1 - \frac{f}{2f+1}\right)\left(1 - \frac{f}{2f}\right) \leq \left(\frac{3}{5}\right)\left(\frac{1}{2}\right) = \frac{3}{10} \quad \text{(recall that } f \geq 2\text{)}$$

Therefore, L.H.S. is upper bounded by $2\left(\frac{3}{10}\right) - 0 = \frac{6}{10} \leq 1 - \frac{1}{40} \leq 1 - \frac{p_{rnd}^2}{40}$ (since $p_{rnd} \leq 1$).

Note that when $2f < c - f + 1$: $\left(\frac{2\binom{c-f}{f} - \binom{c-2f}{f}}{\binom{c}{f}} \right) = 2\prod_{i=1}^{f}\left(1 - \frac{f}{c-i+1}\right) - \prod_{i=1}^{f}\left(1 - \frac{2f}{c-i+1}\right)$.

The next two cases pertain to this regime.

Case 3: $f < \frac{c-f+1}{2}$ and $\sum_{i=1}^{f} \frac{f}{c-i+1} > 0.8$

Set $x_i = \frac{f}{c-i+1}$. Note that $1 - p_{rnd} = \prod_{i=1}^{f}(1 - x_i) \leq \prod_{i=1}^{f} e^{-x_i} = e^{-\sum x_i} < e^{-0.8} \leq 0.45$. Therefore $p_{rnd} \geq 0.55$. Hence:

$$2\prod_{i=1}^{f}(1 - x_i) - \prod_{i=1}^{f}(1 - 2x_i) \leq 2\prod_{i=1}^{f}(1 - x_i) \leq 0.9 \leq 1 - \frac{p_{rnd}}{10} \leq 1 - \frac{p_{rnd}^2}{10} \leq 1 - \frac{p_{rnd}^2}{40} \quad (4.8)$$

Case 4: $f < \frac{c-f+1}{2}$ and $\sum_{i=1}^{f} \frac{f}{c-i+1} \leq 0.8$

42

Denote by $I_{2m+1\leq f}$ the indicator variable which is 1 if $2m+1 \leq f$ and 0 else. We first prove the following inequality:

$$2\prod_{i=1}^{f}(1-x_i) - \prod_{i=1}^{f}(1-2x_i)$$

$$= \left(2 - 2\sum_i x_i + 2\frac{\sum_i (x_i \sum_{j\neq i} x_j)}{2!} - 2\frac{\sum_i (x_i \sum_{j\neq i}(x_j \sum_{k\neq i,j}(x_k)))}{3!} + ... + 2(-1)^f x_1 x_2 ... x_f\right)$$

$$- \left(1 - \sum_i (2x_i) + \frac{\sum_i ((2x_i)\sum_{j\neq i}(2x_j))}{2!} + ... + (-1)^f (2x_1)(2x_2)...(2x_f)\right)$$

$$= 1 + (2-2^2)\frac{\sum_i (x_i \sum_{j\neq i}(x_j))}{2!} - (2-2^3)\frac{\sum_i (x_i \sum_{j\neq i}(x_j \sum_{k\neq i,j} x_k))}{3!} + ... + (-1)^f (2-2^f) x_1 x_2 ... x_f$$

$$= 1 - \sum_i \left(x_i \left(\sum_{j\neq i} x_j\right)\right) + \sum_i \left(x_i \sum_{j\neq i}\left(x_j \sum_{k\neq i,j} x_k\right)\right)$$

$$+ \sum_{m=2}^{\lfloor \frac{f}{2} \rfloor} \left[\frac{(2-2^{2m})}{(2m)!}\sum_{i_1}\left(x_{i_1}\left(\sum_{i_2\neq i_1}\left(x_{i_2}... \sum_{i_{2m}\neq i_1,i_2,...} x_{i_{2m}}\right)\right)\right)\right.$$

$$\left. - \frac{(2-2^{2m+1})}{(2m+1)!}\sum_{i_1}\left(x_{i_1}\sum_{i_2\neq i_1}\left(x_{i_2}... \sum_{i_{2m+1}\neq i_1,i_2,...} x_{i_{2m+1}}\right)\right)I_{2m+1\leq f}\right]$$

$$= 1 - \sum_i \left(x_i \left(\sum_{j\neq i} x_j\right)\right) + \sum_i x_i \left(\sum_{j\neq i} x_j \left(\sum_{k\neq i,j} x_k\right)\right)$$

$$+ \sum_{m=2}^{\lfloor \frac{f}{2} \rfloor} \left[\frac{(2-2^{2m})}{(2m)!}\sum_{i_1}\left(x_{i_1}\sum_{i_2\neq i_1} x_{i_2} \left(... \sum_{i_{2m}\neq i_1,i_2,...} x_{i_{2m}}\left(1\right.\right.\right.\right.$$

$$\left.\left.\left.\left. - \frac{(2-2^{2m+1})}{(2m+1)(2-2^{2m})}\sum_{i_{2m+1}\neq i_1,i_2,...} x_{i_{2m+1}}I_{2m+1\leq f}\right)\right)\right)\right]$$

$$= 1 - 0.2\sum_i \left(x_i \sum_{j\neq i} x_j\right) - \left(0.8\sum_i x_i \left(\sum_{j\neq i} x_j\right)\right) - \sum_i \left(x_i \sum_{j\neq i}\left(x_j \left(\sum_{k\neq i,j} x_k\right)\right)\right)$$

$$- \sum_{m=2}^{\lfloor \frac{f}{2} \rfloor} \left[\frac{(2^{2m}-2)}{(2m)!}\sum_{i_1}\left(x_{i_1}\sum_{i_2\neq i_1} x_{i_2} \left(... \sum_{i_{2m}\neq i_1,i_2,...} x_{i_{2m}}\left(1\right.\right.\right.\right.$$

$$\left.\left.\left.\left. - \frac{(2^{2m+1}-2)}{(2m+1)(2^{2m}-2)}\sum_{i_{2m+1}\neq i_1,i_2,...} x_{i_{2m+1}}I_{2m+1\leq f}\right)\right)\right)\right]$$

$$\leq 1 - 0.2 \sum_i \left(x_i \sum_{j \neq i} x_j \right) \text{ whenever } \sum_i x_i \leq 0.8$$

$$\because 1 - \frac{(2^{2m+1} - 2)}{(2m+1)(2^{2m} - 2)} \sum_{i_{2m+1} \neq i_1, i_2, \ldots} x_{i_{2m+1}} I_{2m+1 \leq f} \geq 0 \; \forall m \geq 2 \text{ when } \sum_i x_i \leq 0.8$$

Set $x_i = \frac{f}{c-i+1}$. By assumption $\sum_i x_i = \sum_{i=1}^{f} \frac{f}{c-f+1} \leq 0.8$. Also $\sum_i (x_i \sum_{j \neq i} x_j) \geq f(f-1)\frac{f^2}{c^2} \geq$

$\frac{1}{2}(\frac{f^2}{c})^2 \geq \frac{1}{2}(\frac{p_{rnd}}{2})^2 = \frac{p_{rnd}^2}{8}$ (applying Lemma 14). Hence $2 \prod_{i=1}^{f} (1 - x_i) - \prod_{i=1}^{f} (1 - 2x_i) \leq$

$1 - 0.2 \sum_i \sum_{j \neq i} x_i x_j \leq 1 - \frac{p_{rnd}^2}{40}$. □

4.4 Conditions for Connectivity

We now show that the critical range for connectivity with random (c, f) assignment in the regime $c = O(\log n)$ is $\Theta(\sqrt{\frac{\log n}{p_{rnd}n}})$.

4.4.1 Necessary Condition for Connectivity

Theorem 4. *With a random (c, f) channel assignment (when $c = O(\log n)$), if $\pi r^2(n) = \frac{(\log n + b(n))}{pn}$, where $p = p_{rnd} = 1 - (1 - \frac{f}{c})(1 - \frac{f}{c-1})...(1 - \frac{f}{c-f+1})$, and $c = O(\log n)$, and $\limsup_{n \to \infty} b(n) = b < +\infty$ then:*

$$\liminf_{n \to \infty} \; \Pr[\text{ disconnection }] \geq e^{-b}(1 - e^{-b})$$

where by disconnection we imply the event that there is a partition of the network.

Proof. The proof is obtained by an adaptation of the proof technique used in [42]. We provide a proof-sketch here. The detailed proof is described in Appendix B.

We focus on the disconnection events where some node(s) are *isolated*.

From the model definition, the probability that two nodes in range of each other can operate on at least one common channel is $p = p_{rnd}$ where $1 - p_{rnd} = (1 - \frac{f}{c})(1 - \frac{f}{c-1})...(1 - \frac{f}{c-f+1})$.

The probability that a node x is isolated, i.e., cannot communicate with any other node, is give by $p_1 = (1 - p\pi r^2(n))^{(n-1)}$. One can also obtain an upper bound p_2 on the probability

that two nodes x and y are both isolated. It can be shown that:

$$
\begin{aligned}
\Pr[\text{ disconnection }] &\geq \sum_x \Pr[x \text{ is only isolated node}] \\
&\geq \sum_x \Pr[x \text{ isolated }] - \sum_{x,y \neq x} \Pr[x \text{ and } y \text{ both isolated }] \\
&\geq np_1 - n(n-1)p_2 \\
&\geq \theta e^{-b} - (1+\epsilon)e^{-2b} \quad \text{where } b = \limsup_{n \to \infty} b(n) \\
&\text{for any } \theta < 1, \epsilon > 0, \text{ and large } n
\end{aligned}
$$

(4.9)

Therefore, if $\limsup_{n \to \infty} b(n) = b < +\infty$, the network is asymptotically disconnected with some positive probability. $\qquad \square$

Corollary 2. *With a random (c, f) assignment, the necessary condition for connectivity is that $r(n) = \Omega(\sqrt{\frac{\log n}{p_{rnd}n}})$, else the network is disconnected with some positive probability.*

4.4.2 Sufficient Condition for Connectivity

Theorem 5. *With random (c, f) assignment, in the regime $c = O(\log n)$, if $\pi r^2(n) = \frac{800\pi \log n}{p_{rnd}n}$, then:*

$$
\lim_{n \to \infty} \Pr[\text{ network is connected }] = 1
$$

Proof. The construction is based on per-node structures termed as *backbones*.

Consider a subdivision of the unit torus into square cells of area $a(n) = \frac{100 \log n}{p_{rnd}n}$. Noting that $p_{rnd} \geq \frac{f}{c} = \Omega(\frac{1}{\log n})$, and setting $\alpha(n) = \frac{1}{p_{rnd}}$ in Lemma 59, there are at least $\frac{50 \log n}{p_{rnd}}$ nodes in each cell with probability at least $1 - \frac{50 \log n}{n}$. Choose $r(n) = \sqrt{8a(n)}$. Resultantly, a node in any given cell has all nodes in adjacent cells within its range.

Within each cell, we categorize nodes as either *transition facilitators* or *backbone candidates* (the meaning of these terms shall become clear later) in the following manner: We choose $\lfloor \frac{2 \log n}{p_{rnd}} \rfloor$ nodes uniformly at random, and set them apart as *transition facilitators*. This leaves at least $\lceil \frac{48 \log n}{p_{rnd}} \rceil$ nodes in each cell that can be deemed as *backbone candidates*.

Consider any node in any given cell. The probability that it can communicate with any other random node in its range is p_{rnd}. Hence, the probability that in an adjacent cell, there

is no backbone candidate node with which it can communicate is at most $(1-p_{rnd})^{\left\lceil \frac{48 \log n}{p_{rnd}} \right\rceil} \leq$ $\frac{1}{e^{48 \log n}} = \frac{1}{n^{48}}$ (applying Fact 2).

The probability that a given node cannot communicate with any node in some adjacent cell is thus at most $\frac{8}{n^{48}}$ (applying the union bound over all 8 possible adjacent cells). By applying the union bound over all n nodes, the probability that at least one node is unable to communicate with any backbone candidate node in at least one of its adjacent cells is at most $\frac{8}{n^{47}}$.

We associate with each node x a set of nodes and links $\mathcal{B}(x)$ called the backbone for x. $\mathcal{B}(x)$ is constructed as follows:

Throughout the procedure, cells that are already covered by the under-construction backbone are referred to as *filled* cells. x is by default a member of $\mathcal{B}(x)$, and its cell is the first *filled* cell. From each adjacent cell, amongst all backbone candidate nodes sharing at least one common channel with x, one node (and hence also the link between that node and x) is chosen uniformly at random and added to $\mathcal{B}(x)$. Thereafter, from each unfilled cell bordering a filled cell, of all nodes sharing at least one common channel (and hence a feasible link) with some node already in $\mathcal{B}(x)$, one is chosen uniformly at random, and is added to $\mathcal{B}(x)$ (the link on the basis of which this node was chosen is added as a backbone link); the cell containing the chosen node gets added to the set of filled cells. This process continues iteratively, till there is one node from every cell in $\mathcal{B}(x)$. From our earlier observations, for all nodes x, $\mathcal{B}(x)$ will eventually cover all cells with probability at least $1 - \frac{8}{n^{47}}$. Note that from any node in $\mathcal{B}(x)$ there is a path to x comprising entirely of links in the backbone.

Now consider any pair of nodes x and y. If there exists a connected path between some node in $\mathcal{B}(x)$ and some node in $\mathcal{B}(y)$ then x and y are connected. This can occur in many different ways. Consider three possibilities (Fig. 4.1.)

If $\mathcal{B}(x)$ and $\mathcal{B}(y)$ have a common node (Fig. 4.1(a)), then the two nodes are obviously connected, as one can proceed from x on $\mathcal{B}(x)$ towards one of the common nodes, and thence to y on $\mathcal{B}(y)$, and vice-versa.

Suppose the two backbones are disjoint. Then x and y are still connected if there is some cell such that the node belonging to $\mathcal{B}(x)$ in that cell (let us call it q_x) can communicate with the node belonging to $\mathcal{B}(y)$ in that cell (let us call it q_y), either directly, or through a third node. q_x and q_y can always communicate directly if they share a common channel
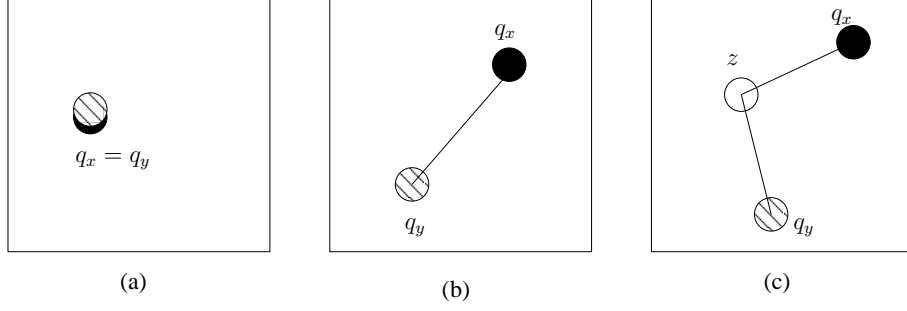
Figure 4.1: Some ways in which backbones can be connected

(Fig. 4.1(b)). Hence, the case of interest is one in which no cell has q_x and q_y sharing a channel.

Consider a particular cell, with q_x and q_y as the respective backbone members. If q_x and q_y do not share a common channel, we consider the event that there exists a third node amongst the *transition facilitators* in the cell through whom they can communicate (Fig. 4.1(c)). Given backbones $\mathcal{B}(x)$ and $\mathcal{B}(y)$, and given a network cell in which q_x and q_y do not share a channel, the probability that they can both communicate with a given third node z that did not participate in backbone formation and is known to lie in the same cell, is independent of the probability of a similar event in another cell.

Therefore, the overall probability can be lower-bounded by obtaining for one cell the probability of q_x and q_y communicating via a third node z in the cell given they have no common channel, taking into account that each cell has at least $\lfloor \frac{2 \log n}{p_{rnd}} \rfloor$ possibilities for z, and treating it as independent across cells. We elaborate on this further:

Let q_x have the set of channels $C(q_x) = \{c_{x_1}, ..., c_{x_f}\}$, and q_y have the set of channels $C(q_y) = \{c_{y_1}, ..., c_{y_f}\}$, such that $C(q_x) \cap C(q_y) = \phi$.

Consider a third node z amongst the transition facilitators in the same cell as q_x and q_y. Denote the set of z's chanels by $C(z)$. We desire that $C(z) \cap C(q_x) \neq \phi$ and $C(z) \cap C(q_y) \neq \phi$.

Note that a node x is a member of its own backbone. Thus $q_x = x$ in x's cell, and if x is a transition facilitator, this would imply that $q_x = x$ is not a backbone candidate. To maintain uniformity and clarity, let us therefore only consider cells other than those in which x and y lie (this can lead to the exclusion of at most 2 cells). In any such cell, q_x and q_y are both backbone candidates, and if they do not share a common channel, it implies that they can communicate through a given transition facilitator z with probability

47

$p_z = 1 - \left(\frac{2\binom{c-f}{f} - \binom{c-2f}{f}}{\binom{c}{f}} \right) \geq \frac{p_{rnd}^2}{40}$ (Lemma 16).

There are $\lfloor \frac{2\log n}{p_{rnd}} \rfloor$ possibilities for z within that cell if neither x and y lie in the cell (since in that case q_x, q_y are both *backbone candidates*), and all the possible z nodes have i.i.d. channel assignments. Thus, the probability that q_x and q_y cannot communicate through any z in the cell is at most $(1 - p_z)^{\lfloor \frac{2\log n}{p_{rnd}} \rfloor}$, and the probability they can indeed do so is $p_{xy} \geq 1 - (1 - p_z)^{\lfloor \frac{2\log n}{p_{rnd}} \rfloor}$.

The number of such cells is at least $\frac{1}{a(n)} - 2 = \frac{p_{rnd}n}{100\log n} - 2$. Therefore, the probability that this happens in none of the $\frac{p_{rnd}n}{100\log n} - 2$ cells is at most $(1 - p_{xy})^{\frac{p_{rnd}n}{100\log n} - 2} \leq (1 - p_z)^{(\lfloor \frac{2\log n}{p_{rnd}} \rfloor)(\frac{p_{rnd}n}{100\log n} - 2)} \leq e^{-(\frac{p_{rnd}^2}{40})(\lfloor \frac{2\log n}{p_{rnd}} \rfloor)(\frac{p_{rnd}n}{100\log n} - 2)} = e^{-\Omega(\frac{n}{\log^2 n})}$ (recall that $c = O(\log n)$ and therefore $p_{rnd} = \Omega(\frac{1}{\log n})$, and of course $p_{rnd} \leq 1$).

Applying the union bound over all $\binom{n}{2} < \frac{n^2}{2}$ node pairs, the probability that some pair of nodes are not connected is at most $\frac{n^2 e^{-\Omega(\frac{n}{\log^2 n})}}{2} \leq \frac{1}{2}e^{-\Omega(\frac{n}{\log^2 n}) + 2\log n}$. Applying another union bound over this probability, the probability that some of the cells are not sufficiently populated (as mentioned earlier, this probability is at most $\frac{50\log n}{n}$), and the probability that some backbone cannot be grown fully (at most $\frac{8}{n^{47}}$), we obtain that the probability of a connected network converges to 1, as $n \to \infty$. $\qquad\square$

4.5 Upper Bound on Capacity

Theorem 4 established that unless $r(n) = \Omega(\sqrt{\frac{\log n}{p_{rnd}n}})$, some node is isolated with positive probability. In Section 2.5 of Chapter 2, we discussed how the need to have $r(n) = \Omega(g(n))$ implies that capacity is constrained to be $O(\frac{W}{ng(n)})$. In light of this, it follows that, for the random (c, f) model in the regime $c = O(\log n)$, the per flow capacity is $O(W\sqrt{\frac{p_{rnd}}{n\log n}})$.

4.6 A Sub-Optimal Lower Bound on Capacity

We describe a construction CR_1 that achieves a per-flow throughput of $\Omega(W\sqrt{\frac{f}{cn\log n}})$. Though it is not optimal, this construction is of interest for the following reasons:

- The optimal procedure uses this construction for $f < 100$.

- This construction involves a simple routing and scheduling procedure, in contrast to the optimal procedure for $f \geq 100$ described in Section 4.7. Thus, it exemplifies a

performance-complexity trade-off.

This construction is quite similar to the construction for adjacent (c, f) assignment. The surface of the unit torus is divided into square cells of appropriate area $a(n)$ each. The transmission range is set to $\sqrt{8a(n)}$, thereby ensuring that any node in a given cell is within range of any other node in any adjoining cell. The number of cells that interfere with a given cell is only some constant (say γ). We choose $a(n) = \frac{100c \log n}{fn}$.

Lemma 17. *There are at least $\frac{50c \log n}{f}$ and at most $\frac{150c \log n}{f}$ nodes in every cell w.h.p.*

Proof. By application of Lemma 59, we can show that the number of nodes in any cell lies between $\frac{50c \log n}{f}$ and $\frac{150c \log n}{f}$ with probability at least $1 - \frac{50 \log n}{n}$. □

Lemma 18. *If there are at least $\frac{50c \log n}{f}$ nodes in every cell, then with probability at least $1 - O(\frac{1}{n^4})$, for each of the c channels, there are at least $25 \log n$ nodes in each cell that can switch on that channel.*

Proof. Let us consider one particular cell $\mathcal{H}$. Let $X_{ij} = 1$ if node j can switch on channel i, and 0 else. $\Pr[X_{ij} = 1] = \frac{f}{c}$, and, for a given i, all the X_{ij}'s are independent. Let $X_i = \sum_{j \in \mathcal{H}} X_{ij}$.Then $E[X_i] \geq 50 \log n$. By application of the Chernoff bound in Lemma 53 (with $\beta = \frac{1}{2}$) , we obtain:

$$\Pr[X_j \leq 25 \log n] \leq \exp(-\frac{50 \log n}{8}) < \frac{1}{n^6} \qquad (4.10)$$

Since there are $c = O(\log n)$ channels, the union bound yields that $\Pr[X_i \leq 25 \log n$ for any $i \in 1, 2, ..., c] \leq \frac{c}{n^6} = O(\frac{\log n}{n^6}) \implies O(\frac{1}{n^5})$. Further, since there are $\frac{1}{a(n)} = \frac{fn}{100c \log n} < n$ cells, another application of the union bound yields:

$$\Pr[\text{ less than } 25 \log n \text{ nodes per channel in any cell}] = O(\frac{1}{n^4}) \qquad (4.11)$$

□

4.6.1 Routing

Initially, each flow is assigned a source channel l, as well as a target destination channel r. The source channel for a flow originating at node S is chosen according to the uniform

distribution from the f channels available at S. The destination channel may be chosen from amongst the f channels available at destination D in any manner.

We need to find a feasible path from S to D. To obtain a feasible path, we try to find a sequence of nodes $v_0 = S, v_1, v_2, ..., \mathbf{v_i}, ..., v_k = D$ such that, for all $0 \leq m \leq i$, v_m can operate on channel l, and for all $i \leq m \leq k$, v_m can operate on r. Thus, node v_i on the route is capable of switching (operating) on both l and r, and this node serves as a *transition point* for the flow's route. To be able to find such a node v_i, we may need to inspect a certain minimum number of cells.

Lemma 19. *If each flow traverses and inspects $h \geq \lceil \frac{2(c-1)}{25(f-1)} \rceil$ distinct cells, where the cells to be inspected are chosen in a manner independent of channel presence in that cell (the cells inspected by any single flow should be distinct; two flows may traverse the same cell), a transition-point (relay node) that can switch on both the flow's source channel, and the flow's destination channel will be found by each flow w.h.p.*

Proof. Consider a particular flow. From Lemma 17, each cell has at least $\frac{50c \log n}{f}$ nodes w.h.p. The probability that there is no node capable of operating on both channels i and j in a given cell along the flow's route is at most $(1 - \frac{f(f-1)}{c(c-1)})^{\frac{50c \log n}{f}}$ (since nodes are assigned channels in an i.i.d. manner). Thus the probability of not finding such a node after h hops is at most $(1 - \frac{f(f-1)}{c(c-1)})^{(\frac{50hc \log n}{f})}$. If $h \geq \lceil \frac{2(c-1)}{25(f-1)} \rceil$, then after traversing h distinct cells, the probability of not finding such a node is at most $(1 - \frac{f(f-1)}{c(c-1)})^{\frac{4c(c-1) \log n}{(f(f-1))}} \leq \exp(-4 \log n) \leq \frac{1}{n^4}$.

Applying the union bound over all n flows, the probability that this should happen for even one flow is at most $\frac{1}{n^3}$. Hence, all flows will have be able to make the required transition w.h.p., after traversing $h \geq \lceil \frac{2(c-1)}{25(f-1)} \rceil$ distinct hops. $\square$

Note that $\frac{2(c-1)}{25(f-1)} \leq \frac{4c}{25f}$. Thus, if we ensure that each flow's route passes through at least $\lceil \frac{4c}{25f} \rceil$ intermediate cells, we will be able to find an end-to-end feasible route for each flow w.h.p. Therefore, we adopt the following routing strategy:

The (almost) straight-line $SD'D$ path is followed if either source and destination channels are the same, or if the straight-line segment SD' provides $h \geq \lceil \frac{4c}{25f} \rceil$ *intermediate* hops. If S and D' (hence also D) lie close to each other, the hop-length of the straight line cell-to-cell path can be much smaller. In this case, a *detour* path is chosen, in a manner similar to that described in Chapter 3 for adjacent (c, f) assignment, and depicted in Fig. 4.4, by considering a circle of radius $\lceil \frac{4c}{f} \rceil r(n)$ centered at S, selecting a point P on the

circumference of that circle, and routing the flow along the sequence of cells traversed by SP, PD', and then a possible additional last hop.

Similar to the construction for adjacent (c, f) assignment described in Chapter 3, we associate two phases with a flow's route: a *progress-on-source-channel* phase, and a *ready-for-transition* phase. We stipulate that a non-detour-routed flow stays in the *progress-on-source-channel* phase along the route, till there are only $\lceil \frac{4c}{25f} \rceil$ intermediate hops left to the destination. At this point, it enters a *ready-for-transition* phase, and is prepared to make a transition given an appropriate relay node that provides the requisite channel-pair for transition (the relay selection strategy is described later). A detour-routed flow is always in *ready-for-transition* phase.

The need to perform *detour* routing for some source-destination pairs does not have any substantial effect on the number of flow-routes that traverse a cell.

Lemma 20. *The number of straight-line $SD'D$ flow-routes that traverse any cell is $O(n\sqrt{a(n)})$.*

Proof. From Lemma 61, the number of SD' straight-lines traversing a single cell are $O(n\sqrt{a(n)})$, yielding $O(n\sqrt{a(n)})$ flow-routes.

We must now separately consider the number of routes whose last $D'D$ hop may enter this cell. If D is in the same cell as D', there is no extra hop. Otherwise, the number of flows for which D' lies in one of the 8 adjacent cells is $O(na(n))$ w.h.p. (since it follows from Lemma 59 (applied to the set of n pseudo-destinations) that the number of pseudo-destinations in any cell is $O(na(n))$). Since $na(n) = O(n\sqrt{a(n)})$, the total number of traversing flow-routes is $O(n\sqrt{a(n)})$. □

Lemma 21. *If the number of flow-routes traversing any cell is x with only straight-line routing, it is $x + O(n \left(\frac{c}{f} \right)^2 r(n)^2) \implies x + O(\log^4 n)$ with detour routing.*

Proof. The detour occurs only when the straight-line route has less than $\lceil \frac{4c}{25f} \rceil$ intermediate hops, and the new route lies entirely within a circle of radius $\lceil \frac{4c}{25f} \rceil r(n)$ around the source. Thus, the extra flows that may pass through a cell (compared to straight-line routing) are only those whose sources lie within a distance $\lceil \frac{4c}{25f} \rceil r(n)$ from some point in this cell. All such possible sources fall within a circle of radius $(1 + \lceil \frac{4c}{25f} \rceil)r(n)$, and hence area $a_c(n) = \pi \left(1 + \lceil \frac{4c}{25f} \rceil \right)^2 r^2(n)$. Noting that the source locations are i.i.d., and applying Lemma 60, any circle of this area has $O(na_c(n))$ nodes, and hence $O(na_c(n))$ sources w.h.p.

Thus, the number of extra flows that traverse any cell due to detour routing is $O(na_c(n))$. Each such flow may traverse a cell at most twice along the SPD' segment, and possibly once more in the additional last hop. Therefore, the total number of flow-routes is $x + O(n \left(\frac{c}{f}\right)^2 r^2(n)) \implies x + O(\log^4 n)$ w.h.p. $\qquad\qquad\square$

Lemma 22. *The number of flow-routes traversing any cell is $O(n\sqrt{a(n)})$ even with detour routing.*

Proof. This follows from Lemma 20, Lemma 21 and the observation that $O(\log^4 n) \implies O(n\sqrt{a(n)})$ for our choice of $a(n)$. $\qquad\qquad\square$

Lemma 23. *The number of flow-routes traversing any cell in* ready-for-transition *phase is $O(\log^4 n)$ w.h.p.*

Proof. We first account for the flows traversing the cell along the SD' segment, and later account for the possible additional $D'D$ hop.

By our construction, a non-detour-routed flow enters the *ready-for-transition* phase only when it is $\Theta(\frac{c}{f})$ hops away from its destination. All such flows must have their pseudo-destinations within a circle of radius $\Theta(\frac{c}{f}r(n))$ centered in the cell. The number of pseudo-destinations that lie within *any circle* of radius $\Theta(\frac{c}{f}r(n))$ from the cell is $\Theta(n\frac{c^2}{f^2}r^2(n)) = O(\frac{c^3}{f^3}\log n) \implies O(\log^4 n)$ w.h.p. (by suitable choice of $\alpha(n)$ in Lemma 60, and by observing that $c = O(\log n)$).

A detour-routed flow is always in *ready-for-transition* phase. From Lemma 21, there are at most $O(\log^4 n)$ such flows, and they can traverse a cell at most twice along the SD' (more precisely SPD') segment, yielding $O(\log^4 n)$ distinct flow-routes.

We now account for the fact that all the above routed flows could have an additional last $D'D$ hop that may need to be counted separately. As argued in the proof of Lemma 20, these yield $O(na(n)) = O(\frac{c\log n}{f}) \implies O(\log^2 n)$ additional traversals.

Hence the number of flow-routes traversing any cell in *ready-for-transition* phase (counting repeat traversals separately) is $O(\log^4 n)$ w.h.p. $\qquad\qquad\square$

Relay Node Selection In the *progress-on-source-channel* phase, the flow's packets are transmitted on the source channel. During this phase, the next hop node is chosen to be the node in the next cell which has the smallest number of flow-links assigned so far for relaying *on that channel*, amongst all nodes that can switch on the source channel.

In the *ready-for-transition* phase, the goal is to seek a relay node that can operate on both the source channel and the destination channel, and therefore is capable of serving as the transition point. It makes use of the first opportunity that presents itself, i.e., if a node in an on-route cell provides the source-destination channel pair, the flow is assigned to that node for relaying (the flow's packets are receievd by the node on the source channel, and transmitted to the next hop node on the destination channel). Once it has made the transition into the destination channel, it remains on that channel. In the *ready-for-transition* phase, it may be assigned to *any eligible node* that provides either the transition opportunity, or the source channel (for flows yet to find a transition), or the destination channel (for flows that have already transitioned into their destination channel).

4.6.2 Load Balance within a Cell

A flow-link is said to *enter* a cell $\mathcal{H}$ on a channel j if the flow's route includes a hop (link) (v_{i-1}, v_i), where v_{i-1} is in a cell adjacent to $\mathcal{H}$, v_i is in $\mathcal{H}$, and v_{i-1} transmits the flow's packets to v_i using channel j (this naturally implies that both v_{i-1} and v_i can operate on channel j). Similarly, a flow-link is said to leave a cell $\mathcal{H}$ on channel j if the route includes a link (v_i, v_{i+1}), where v_i is in $\mathcal{H}$, v_{i+1} is in a cell adjacent to $\mathcal{H}$, and v_i transmits the flow's packets to v_{i+1} using channel j.

Per-Channel Load Recall that each cell has $O(na(n))$ nodes w.h.p., and $O(n\sqrt{a(n)})$ flows traversing it w.h.p.

Lemma 24. *The number of flow-links that enter any cell on any single channel is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p.*

Proof. Consider a particular cell $\mathcal{H}$. A flow-link may enter the cell on channel i if:

1. The flow's source channel is i and it is in *progress-on-source-channel* phase

2. The flow is in *ready-for-transition* phase, its source channel is i, but is yet to find a transition into the destination channel

3. The flow is in *ready-for-transition* phase, its destination channel is i, and it has already made a transition

Recall that the sequence of cells traversed by a flow was chosen in a manner that did not depend on the channels the source node can switch on. Since each node's interface is assigned a random subset of f channels, and it further makes an i.i.d. choice of a source channel from amongst these, it follows that a flow's source channel can be any of $1, 2, ..., c$ with equal probability. Furthermore, the source channels for different flows are independent. However, the destination channels of flows are not necessarily independent, since two flows with the same destination are more likely to have the same destination channel.

Thus, if a flow-link enters the cell in *progress-on-source-channel* phase (also referred to as a non-transitioning flow-link), it is equally likely to be on any channel:

$$\Pr[\text{ flow-link is on channel } i] = \frac{1}{c}, \ \forall 1 \leq i \leq c$$

Denote the number of flow-links entering the cell in *progress-on-source-channel* phase by m. From Lemma 20 and Lemma 22, it follows that $m = O(n\sqrt{a(n)})$.

Let X_{ij} be an indicator variable which is 1 if flow-link j enters the cell on channel i, and is 0 else.

Then $X_i = \sum_j X_{ij}$ denotes the number of flow-links in *progress-on-source-channel* phase that enter the cell on channel i, and $E[X_i] = \frac{m}{c}$. The X_{ij}'s are i.i.d. random variables for a given i, as each flow's source channel is chosen in an i.i.d. manner (though they may not be independent for different i, since $X_{ij} = 1 \implies X_{kj} = 0 \ \forall k \neq i$). Hence, we may set $(1+\beta)E[X_i] = \max\{\frac{e^2 m}{c}, 3\log n\}$ (note that $\beta \geq e^2 - 1 > 0$) apply the Chernoff bound from Lemma 51, and obtain that:

$$\begin{aligned}
\Pr[X_i &\geq \max\{\frac{e^2 m}{c}, 3\log n\}] \\
&\leq \left(\frac{e^\beta}{(1+\beta)^{(1+\beta)}}\right)^{E[X_i]} \\
&\leq \left(\frac{e}{(1+\beta)}\right)^{(1+\beta)E[X_i]} \\
&\leq \left(\frac{eE[X_i]}{\max\{\frac{e^2 m}{c}, 3\log n\}}\right)^{(1+\beta)E[X_i]} \\
&\leq \exp(-(1+\beta)E[X_i]) \\
&\leq \exp(-\max\{\frac{e^2 m}{c}, 3\log n\})
\end{aligned} \tag{4.12}$$

Taking the union bound over all c channels, the probability that any channel has more than $\max\{\frac{e^2 m}{c}, 3\log n\}$ flows is at most $c\exp(-\max\{\frac{e^2 m}{c}, 3\log n\})$. Taking another union bound over all $\frac{1}{a(n)} = \frac{fn}{100c\log n}$ cells, this probability is at most $\frac{fn}{100\log n}\exp(-\max\{\frac{e^2 m}{c}, 3\log n\}) = O(\frac{1}{n^2})$.

Since $\max\{\frac{e^2 m}{c}, 3\log n\} = O(\frac{n\sqrt{a(n)}}{c})$ (note that m is $O(n\sqrt{a(n)})$ and $\log n$ is $O(\frac{n\sqrt{a(n)}}{c})$), we have proved that the number of flow-links that enter any cell in *progress-on-source-channel* phase on any single channel is $O(\frac{n\sqrt{a(n)}}{c})$.

We now account for the flow-links that enter a cell in their *ready-for-transition* phase.

From Lemma 23 there are $O(\log^4 n)$ flow-routes traversing any cell in this phase w.h.p. (counting repeat traversals separately). Thus, the additional overhead posed by the corresponding flow-links on any channel is $O(\log^4 n)$ w.h.p.

Hence, the per-channel load in each cell is at most $O(\frac{n\sqrt{a(n)}}{c}) + O(\log^4 n) \implies O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. $\qquad\square$

Lemma 25. *The number of flow-links that leave any cell on any single channel is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p.*

The proof follows by taking note of Lemma 24, and then applying the same argument as that for Lemma 12.

Per-Node Load

Lemma 26. *The number of flow-links that are assigned to any one node in any cell is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p.*

Proof. A node is always assigned an outgoing link for the single flow for which it is the source. A node is also assigned an incoming flow-link for flows for which it is the destination (these flows terminate in that cell), and there are $O(\log n)$ such flows for any node w.h.p. (Lemma 1).

In addition, a node may be assigned flow-links as a relay on the routes of other flows (for each such route, it is assigned an incoming link as well as an outgoing link).

Some of these flows may be in the *ready-for-transition* phase: for these flows it may provide the required channel pair to facilitate a transition, or provide the source channel (flows yet to find a transition) or destination channel (flows that have already transitioned).

55

From Lemma 23, there are $O(\log^4 n)$ such flow-routes traversing the cell w.h.p. Thus, a node or channel can only have $O(\log^4 n)$ such flow-links assigned for relaying.

It may also be assigned as a relay on the routes of flows that are in *progress-on-source-channel* phase, and do not originate in the cell. We have already established in Lemma 24, that the number of flow-links that enter on a given channel in any cell is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. By construction, we have chosen cell sizes such that there are at least $25\log n$ nodes on each channel in each cell w.h.p. (Lemma 18). Also $c = O(\log n)$. A flow-link in *progress-on-source-channel* phase is always assigned to the node with least load on that channel so far (from amongst all nodes in that cell capable of switching on that channel). From Lemma 24, and the fact that each node can switch on only f channels, the number of such flows that are assigned to any one node is $O(\frac{fn\sqrt{a(n)}}{c\log n}) \implies O(\frac{n\sqrt{a(n)}}{c})$ w.h.p.

The resultant number of assigned flow-links per node is $1 + O(\log n) + O(\log^4 n) + O(\frac{n\sqrt{a(n)}}{c}) \implies O(\frac{n\sqrt{a(n)}}{c})$. $\qquad\square$

4.6.3 Transmission Schedule

The transmission schedule is obtained in a manner similar to the procedure in Section 3.5.3 of Chapter 3, by first obtaining a global inter-cell schedule (recall that the cell-interference graph has chromatic number at most $1 + \gamma$, where γ is a constant independent of n), and then constructing a conflict graph for intra-cell scheduling. From Lemmas 25 and 26, the degree of the conflict graph is $O(\frac{n\sqrt{a(n)}}{c}) + O(\frac{n\sqrt{a(n)}}{c}) = O(\frac{n\sqrt{a(n)}}{c})$. It is well-known that a graph with maximum node degree d has chromatic number at most $d+1$, and so the graph can be colored using $O(\frac{n\sqrt{a(n)}}{c})$ colors.

Thus, the cell-slot is divided into $O(\frac{n\sqrt{a(n)}}{c}) = O(\frac{\sqrt{\frac{cn\log n}{f}}}{c})$ equal length subslots, and each outgoing flow-link gets assigned a slot for transmission on its assigned channel at the per-channel rate of $\frac{W}{c}$ (the slot-assignment is obtained via the conflict-graph coloring described earlier). This yields that each flow will get $\Omega(W\sqrt{\frac{f}{cn\log n}})$ throughput.

In light of the above, we obtain the following theorem:

Theorem 6. *With a random (c, f) channel assignment, when $c = O(\log n)$, construction CR_1 achieves throughput of $\Omega(W\sqrt{\frac{f}{cn\log n}})$ for each flow.*

4.7 Optimal Lower Bound on Capacity

In this section, we present a construction CR^* that achieves $\Omega(W\sqrt{\frac{p_{rnd}}{n\log n}})$ throughput for each flow. In light of the upper bound of $O(W\sqrt{\frac{p_{rnd}}{n\log n}})$ established in Section 4.5, CR^* is optimal for the regime $c = O(\log n)$. This establishes the capacity with random (c, f) assignment as $\Theta(W\sqrt{\frac{p_{rnd}}{n\log n}})$ in the regime $c = O(\log n)$.

We first present a construction CR_2 that achieves $\Omega(W\sqrt{\frac{p_{rnd}}{n\log n}})$ when $f \geq 100$ (thus necessarily $c \geq 100$).

We now describe construction CR_2.

Subdivision of network region into cells Similar to previous constructions, the surface of the unit torus is divided into square cells of area $a(n)$ each, and the transmission range is set to $\sqrt{8a(n)}$, thereby ensuring that any node in a given cell is within range of any other node in any adjoining cell.

We choose $a(n) = \frac{250 \max\{\log n, c\}}{p_{rnd} n} = \Theta(\frac{\log n}{p_{rnd} n})$ (since $c = O(\log n)$).

Lemma 27. *Each cell has at least $\frac{4na(n)}{5} = \frac{200 \max\{\log n, c\}}{p_{rnd}}$ and at most $\frac{6na(n)}{5} = \frac{300 \max\{\log n, c\}}{p_{rnd}}$ nodes w.h.p.*

Proof. We have chosen $a(n) = \frac{250 \max\{\log n, c\}}{p_{rnd} n}$. Thus $a(n) \geq \frac{100 \log n}{p_{rnd} n}$. If $c \leq \log n$, we can set $\alpha = \frac{2.5}{p_{rnd}} > 1$ in Lemma 59, and when $c > \log n$, i.e., $c = \kappa \log n$ (for some $\kappa > 1$) (recall that $c = O(\log n)$), we can set $\alpha = \frac{2.5\kappa}{p_{rnd}} > 1$ (noting that in either case $\alpha \leq \frac{n}{100 \log n}$ for large enough n), to obtain that the following holds with probability at least $1 - \frac{50 \log n}{n}$ for all cells $\mathcal{H}$:

$$\frac{250 \max\{\log n, c\}}{p_{rnd}} - 50 \log n \leq \mathrm{Pop}(\mathcal{H}) \leq \frac{250 \max\{\log n, c\}}{p_{rnd}} + 50 \log n$$

where $\mathrm{Pop}(\mathcal{H})$ denotes the number of nodes in cell $\mathcal{H}$.

Thereafter, noting that $\frac{250 \max\{\log n, c\}}{p_{rnd}} - 50 \log n \geq \frac{200 \max\{\log n, c\}}{p_{rnd}}$, and $\frac{250 \max\{\log n, c\}}{p_{rnd}} + 50 \log n \leq \frac{300 \max\{\log n, c\}}{p_{rnd}}$, completes the proof. $\square$

The following facts will also be used extensively in subsequent proofs:

$$\frac{f}{c} \leq p_{rnd} \leq 1 \tag{4.13}$$

For large n, since $c = O(\log n)$, and $2 \le f \le c$:

$$na(n) = \frac{250 \max\{\log n, c\}}{p_{rnd}} = O(\log^2 n)$$

$$\frac{n\sqrt{a(n)}}{c} = \frac{1}{c}\sqrt{\frac{250n \max\{\log n, c\}}{p_{rnd}}} = \Omega(\sqrt{\frac{n}{\log n}}) \tag{4.14}$$

$$\therefore g(n) = O(na(n)) \implies g(n) = O(\frac{n\sqrt{a(n)}}{c})$$

$$\frac{1}{\sqrt{a(n)}} = \sqrt{\frac{p_{rnd}n}{250 \max\{\log n, c\}}} = O(\sqrt{\frac{n}{\log n}})$$

$$\frac{n\sqrt{a(n)}}{c} = \frac{1}{c}\sqrt{\frac{250n \max\{\log n, c\}}{p_{rnd}}} = \Omega(\sqrt{\frac{n}{\log n}}) \tag{4.15}$$

$$\therefore g(n) = O(\frac{1}{\sqrt{a(n)}}) \implies g(n) = O(\frac{n\sqrt{a(n)}}{c})$$

Some properties of $SD'D$ routing Recall that we use the traffic model of [43], where each source S first chooses a pseudo-destination D', and then selects the node D nearest to it as the actual destination. In [43], the flow traversed cells intersected by the straight line SD', and then took an extra last hop if required (we refer to this as $SD'D$ routing). As we will show later, it may not always suffice to use $SD'D$ routing. However, this is still an important component of our routing procedure. We state and prove certain relevant properties:

Lemma 28. *Given only straight-line SD' routing (no additional last-hop), the number of flows that enter any cell on their i-th hop is at most $\lfloor \frac{5na(n)}{4} \rfloor$ w.h.p., for any i.*

Proof. Let us consider the straight-line part SD' of an $SD'D$ route. All the n SD' lines are i.i.d. Denote by X_i^k the indicator variable which is 1 if the flow k enters a cell $\mathcal{H}$ on its i-th hop. Then, as observed in [36] (proof of Lemma 3 in [36]), for i.i.d. straight lines, the X_i^k's are identically distributed, and X_i^k and X_j^l are independent for $k \ne l$. However, for a given flow k, at most one of the X_i^k's can be 1 as a flow-route only traverses a cell once along the straight line SD'. Then $\Pr[X_i^k = 1] = a(n) = \frac{250 \max\{\log n, c\}}{p_{rnd}n}$.

Let $X_i = \sum_{k=1}^{n} X_i^k$. Then $E[X_i] = na(n)$. Also, for a given i, the X_i^k's are independent

[36]. Then by application of the Chernoff bound from Lemma 52 (with $\beta = \frac{1}{4}$):

$$\Pr[X_i \geq \frac{5E[X_i]}{4}] \leq \exp(-\frac{E[X_i]}{48})$$

$$\therefore \Pr[X_i \geq \frac{1250 \max\{\log n, c\}}{4p_{rnd}}] \leq \exp(-\frac{250 \max\{\log n, c\}}{48p_{rnd}}) < \frac{1}{n^5} \tag{4.16}$$

The maximum value that i can take is $\frac{2}{\sqrt{a(n)}} = \sqrt{\frac{2np_{rnd}}{250 \max\{\log n, c\}}} < n$. Also the number of cells is $\frac{1}{a(n)} \leq n$. By application of union bound over all i, and all cells $\mathcal{H}$, the probability that $X_i \geq \frac{5E[X_i]}{4}$ is less than $\frac{1}{n^3}$, and hence, the number of flows that enter any cell on any hop is less than $\frac{5na(n)}{4} = \frac{1250 \max\{\log n, c\}}{4p_{rnd}}$ with probability at least $1 - \frac{1}{n^3}$. Since X_i is an integer, this implies that it is at most $\lfloor \frac{5na(n)}{4} \rfloor$ w.h.p. $\square$

Lemma 29. *If a node is destination of some flow, that flow's pseudo-destination must lie within either the same cell, or an adjacent cell w.h.p.*

Proof. It was shown in the proof of Lemma 1 that a flow will be assigned to a destination lying within a circle of radius $\sqrt{\frac{100 \log n}{\pi n}}$ centered around the pseudo-destination w.h.p. Conversely, if a flow is assigned to a node, then the pseudo-destination must lie within a circle of of radius $\sqrt{\frac{100 \log n}{\pi n}}$ centered around the node.

It is easy to see that a circle of radius $\sqrt{\frac{100 \log n}{\pi n}}$ centered at a node will fall completely within the cells adjacent to the node's cell (by our choice of cell-area $a(n)$). Hence, if a node is destination of some flow, that flow's pseudo-destination must lie within either the same cell, or an adjacent cell. $\square$

Lemma 30. *The number of $SD'D$ routes that traverse any cell is $O(n\sqrt{a(n)})$ w.h.p.*

Proof. Consider a cell $\mathcal{H}$. From Lemma 61 (which is obtained from a lemma in [36]), we know that the number of SD' straight-lines traversing any single cell are $O(n\sqrt{a(n)})$. We must now consider the number of routes whose last $D'D$ hop may enter this cell $\mathcal{H}$. If D is in the same cell as D', there is no extra hop. Let us now consider the case that D' lies in one of the 8 adjacent cells, but D lies in the cell $\mathcal{H}$ (from Lemma 29, we know that D lies in cell $\mathcal{H}$ only if D' lies in $\mathcal{H}$ or its adjacent cells). The number of flows for which D' lies in one of the 8 cells adjacent to $\mathcal{H}$ is $O(na(n))$ w.h.p. (by applying Lemma 59 to the set of n pseudo-destinations). Also from (4.14), and the fact that $c > 1$, we

know that $O(na(n)) \implies O(n\sqrt{a(n)})$. Therefore, the total number of traversing routes is $O(n\sqrt{a(n)})$. □

Having stated and proved these preliminary lemmas, we now establish some properties of the spatial distribution of channels, and thereafter describe our scheduling/routing procedure:

Definition 2. *(Usability Threshold for Channel Use) The usability threshold for channel use is denoted by M_u, and $M_u = \lceil \frac{9fna(n)}{25c} \rceil = \lceil \frac{90f \max\{\log n, c\}}{cp_{rnd}} \rceil$.*

Lemma 31. *If there are at least $\frac{200 \max\{\log n, c\}}{p_{rnd}}$ nodes in every cell, of which we choose $\frac{180 \max\{\log n, c\}}{p_{rnd}}$ nodes uniformly at random as* candidates *to examine, then, in each cell, amongst those $\frac{180 \max\{\log n, c\}}{p_{rnd}}$ candidate* nodes, *at least $c - \lfloor \frac{f}{4} \rfloor$ channels have at least M_u candidate nodes capable of switching on them, w.h.p.*

Proof. Consider any single cell $\mathcal{H}$. Denote by $\mathcal{S}$ the set of $\frac{180 \max\{\log n, c\}}{p_{rnd}}$ nodes lying in cell $\mathcal{H}$ that are chosen uniformly at random for examination. Denote by I_{ji} the indicator variable that is 1 if a node j can switch on channel i and 0 else. Then: $\Pr[I_{ji} = 1] = \frac{f}{c}$ and for a given i, the I_{ji} are independent. $X_i = \sum_{j \in \mathcal{S}} I_{ji}$ is the number of nodes in $\mathcal{S}$ capable of switching on channel i. Then $E[X_i] = \frac{f}{c} \left(\frac{180 \max\{\log n, c\}}{p_{rnd}} \right)$, and we can see that $M_u = \lceil \frac{E[X_i]}{2} \rceil$.

In light of Lemma 14, we obtain the following:

$$E[X_i] = \frac{180f \max\{\log n, c\}}{cp_{rnd}} \tag{4.17}$$

$$E[X_i] \geq \frac{180 \max\{\log n, c\}}{\min\{2f, \frac{c}{f}\}} \geq \frac{90 \max\{\log n, c\}}{f} \tag{4.18}$$

$$E[X_i] \geq 180f \text{ from (4.17) (noting that } p_{rnd} \leq 1) \tag{4.19}$$

$$E[X_i] \geq \frac{180 \max\{\log n, c\}}{\min\{2f, \frac{c}{f}\}} \geq \frac{180 \max\{\log n, c\}}{\sqrt{2c}} > 90 \max\{\frac{\log n}{\sqrt{c}}, \sqrt{c}\} \geq 90\sqrt{\log n} \tag{4.20}$$

(by applying Lemma 15)

From the preceding equations, it also follows that:

$$M_u \geq \left\lceil \max\{\frac{45 \max\{\log n, c\}}{f}, 90f, 45\sqrt{\log n}\} \right\rceil$$

60

Let I_i' denote an indicator variable which is 1 if $X_i < \frac{E[X_i]}{2}$, and 0 else. Applying the Chernoff bound in Lemma 53:

$$\Pr[I_i' = 1] = \Pr[X_i < \frac{E[X_i]}{2}] \leq \Pr[X_i \leq \frac{E[X_i]}{2}] \leq \exp(-\frac{E[X_i]}{8}) \tag{4.21}$$

Besides, the I_i'''s are negatively correlated, as each node has a uniformly random subset of f channels assigned to it, and thus, in the given set of nodes $\mathcal{S}$, having some channel (say i) assigned to a large number of nodes can only decrease the presence of another channel (say k).

Let $X = \sum_{i=1}^{c} I_i'$. Then, noting[2] that $\log c \leq \frac{E[X_i]}{200}$ $\forall c \geq 2$:

$$E[X] \leq c \exp(-\frac{E[X_i]}{8}) = \exp(-\frac{E[X_i]}{8} + \log c) \leq \exp(-\frac{3E[X_i]}{25})$$
$$(\because \log c \leq \frac{E[X_i]}{200} \ \ \forall c \geq 2) \tag{4.22}$$

Due to the negative correlation of I_i'''s, we can still apply the Chernoff bound (see Lemma 55). By setting $(1+\beta)E[X] = \frac{f}{4}$ in Lemma 51 (from (4.22) $E[X] \leq \exp(-\frac{3E[X_i]}{25}) \leq \exp(-\frac{3}{25}(180f)) < \frac{f}{4}$, yielding $\beta > 0$), we obtain by appropriate substitutions at each step, the following:

$$\Pr[X \geq \lceil \frac{f}{4} \rceil] \leq \Pr[X \geq \frac{f}{4}] \leq \left(\frac{e^\beta}{(1+\beta)^{1+\beta}} \right)^{E[X]} < \left(\frac{e}{1+\beta} \right)^{(1+\beta)E[X]}$$

$$= \left(\frac{4eE[X]}{f} \right)^{\frac{f}{4}}$$

$$\leq \left(\frac{4e \exp(-\frac{3}{25} \left(\frac{90 \max\{\log n, c\}}{f} \right))}{f} \right)^{\frac{f}{4}} \quad \text{from (4.18) and (4.22)}$$

$$= \left(\frac{4e \exp(-\frac{270 \max\{\log n, c\}}{25f})}{f} \right)^{\frac{f}{4}}$$

$$= \frac{\exp\left(\frac{-270 \max\{\log n, c\}}{100} \right)}{\left(\frac{f}{4e} \right)^{\frac{f}{4}}}$$

$$\leq \frac{\exp\left(-2.7 \max\{\log n, c\} \right)}{\left(\frac{1}{2e} \right)^{\frac{f}{4}}}$$

[2]From (4.20): $\frac{E[X_i]}{200} \geq \frac{180 \max\{\log n, c\}}{200\sqrt{2c}} \geq \frac{9 \max\{\log n, c\}}{10\sqrt{2c}} \geq \frac{9\sqrt{c}}{10\sqrt{2}} \geq \log c$ whenever $c \geq 2$.

$$\leq \frac{\exp\left(-2.7\max\{\log n, c\}\right)}{\left(\frac{1}{e^2}\right)^{\frac{f}{4}}} \quad (\text{ since } f \geq 2)$$

$$\leq \exp(-2.7\max\{\log n, c\})\exp(\frac{f}{2}) \quad\quad (4.23)$$

$$\leq \exp(-2\max\{\log n, c\}) \leq \frac{1}{n^2} \quad (\text{ since } f \leq c)$$

Applying the union bound over all $\frac{1}{a(n)} \leq n$ cells in the network, the probability that this happens in any cell is at most $\frac{1}{n}$. Thus, with probability at least $1 - \frac{1}{n}$, $X < \lceil\frac{f}{4}\rceil$, i.e., $X \leq \lfloor\frac{f}{4}\rfloor$ (since X is an integer). Hence, each cell has at least $c - \lfloor\frac{f}{4}\rfloor$ channels with $X_i \geq \frac{E[X_i]}{2}$ *candidate* nodes capable of switching on them. Therefore, from the definition of X, each cell has at least $c - \lfloor\frac{f}{4}\rfloor$ channels with $X_i \geq \lceil\frac{E[X_i]}{2}\rceil$ *candidate* nodes capable of switching on them (since X_i is also an integer). From (4.17) and the definition of M_u, we know that $M_u = \lceil\frac{E[X_i]}{2}\rceil$. This proves the result. $\qquad\square$

Similar to the proof of Theorem 5, the approach involves constructing a routing structure (backbone) for each node. However, in this case, we only need to construct routing structures that can provide a route between the n chosen SD pairs, and not all node pairs. Thus, the constructed backbones are *partial backbones* in that, unlike the proof of Theorem 5, they do not cover all cells in the network. *Moreover, since our concern is not merely connectivity but also capacity, these partial backbones need to be constructed carefully, to ensure that no bottlenecks are formed.*

Similar to the proof of Theorem 5, we begin by classifying all nodes as either *backbone candidates* or *transition facilitators*.

Conditioning on Lemma 27, there are at least $\frac{200\max\{\log n, c\}}{p_{rnd}}$ nodes in each cell w.h.p. Initially, in each cell, we choose $\frac{180\max\{\log n, c\}}{p_{rnd}}$ nodes uniformly at random as *backbone candidates*. The remaining nodes (which are at least $\frac{20\max\{\log n, c\}}{p_{rnd}}$ in number) are deemed *transition facilitators*.[3]

We next define a notion of a channel being *proper* in a cell:

---

[3]The number of nodes in either category must be an integer. Here, for simplicity we assume that we can indeed select exactly $\frac{180\max\{\log n, c\}}{p_{rnd}}$ as backbone candidates and the remaining nodes are at least $\frac{20\max\{\log n, c\}}{p_{rnd}}$. If these two quantities are not integers, but one can select at least $\lceil\frac{180\max\{\log n, c\}}{p_{rnd}}\rceil$ backbone candidates and still have at least $\lceil\frac{20\max\{\log n, c\}}{p_{rnd}}\rceil$ nodes left as transition facilitators, the results will evidently continue to hold. It is also possible to conceive of a scenario where there are exactly $\frac{200\max\{\log n, c\}}{p_{rnd}}$ nodes in the cell, but $\frac{180\max\{\log n, c\}}{p_{rnd}}$ and $\frac{20\max\{\log n, c\}}{p_{rnd}}$ are not integers. In such a scenario, one can select $\lceil\frac{180\max\{\log n, c\}}{p_{rnd}}\rceil$ nodes as backbone candidates and $\lfloor\frac{20\max\{\log n, c\}}{p_{rnd}}\rfloor$ as transition facilitators, without affecting the results (except for a minor change in the probability calculations involving transition facilitators).

Definition 3. (Proper Channel) *A channel i is deemed* proper *in cell $\mathcal{H}$ if at least M_u backbone candidate nodes in $\mathcal{H}$ are capable of switching (operating) on it.*

Note that being proper is a property defined with respect to a specific cell, i.e., a channel can be proper in one cell and not proper in another.

Lemma 32. *For each cell of the network, the following is true w.h.p.: if the number of proper channels in the cell is c', then $c' \geq c - \lfloor \frac{f}{4} \rfloor \geq c - \lfloor \frac{c}{4} \rfloor \geq \lceil \frac{3c}{4} \rceil \geq \frac{3c}{4}$.*

Proof. The proof follows from Lemma 27 and Lemma 31. □

We now prove a property that plays an important role in proving that traffic load can be distributed without creating bottlenecks:

Lemma 33. [4]

Consider a cell $\mathcal{H}$. Let $\mathcal{W}_i$ be the set of all nodes in the 8 adjacent cells $\mathcal{H}(k), 1 \leq k \leq 8$, that are capable of switching on channel i.

For a set of nodes $\mathcal{B}$, define $\mathcal{C}_{\mathcal{H}}(\mathcal{B})$ as:

$$\mathcal{C}_{\mathcal{H}}(\mathcal{B}) = \{j | j \text{ proper in } \mathcal{H} \text{ and } \exists u \in \mathcal{B} \text{ capable of switching on } j\}$$

If $f \geq 100$, the following holds w.h.p.:

$$\forall \mathcal{H}, \ \forall \text{ channels } i, \ \forall \mathcal{B} \subseteq \mathcal{W}_i \text{ such that } |\mathcal{B}| = \lceil \frac{fna(n)}{4c} \rceil : \ |\mathcal{C}_{\mathcal{H}}(\mathcal{B})| \geq \lceil \frac{3c}{8} \rceil$$

Proof. We condition on the node-locations, and their conforming to the high-probability event of Lemma 27. Consider a cell $\mathcal{H}$. Let c' be the number of proper channels in $\mathcal{H}$.

Having conditioned on (and thus fixed) the node-locations (and thereby node-population in each cell), channel-presence in each cell is independent of other cells, as channel assignment is done independently for each node.

Then we can show that: $c' \geq c - \lfloor \frac{f}{4} \rfloor \geq c - \lfloor \frac{c}{4} \rfloor \geq \lceil \frac{3c}{4} \rceil \geq \frac{3c}{4}$, with probability at least $1 - \frac{1}{n^2}$, by following the proof argument of Lemma 31 up to (4.23) (just prior to application of the union bound over all cells in the proof of that lemma).

---

[4]This can be viewed as a special variant of the Coupon Collector's problem [83], where there are c different types of coupons, and each box has a random subset of f different coupons. Some other somewhat different variants having multiple coupons per box have been considered in work on coding, e.g., [33].

If $c' < \frac{3c}{4}$, then we assume that our desired event does not happen for the purpose of obtaining a bound. This probability is at most $\frac{1}{n^2}$.

We now focus on the case where $c' \geq \frac{3c}{4}$.

Consider a particular channel i.

Recall that $\mathcal{W}_i$ is the set of nodes in the cells adjacent to $\mathcal{H}$ that are capable of switching on channel i.

We first bound the probability that $|\mathcal{W}_i| \geq 2400e^2 \max\{\log n, c\}$.

Let Y_{ij} be an indicator variable that is 1 if node j in cells adjacent to $\mathcal{H}$ is capable of switching on channel i, and 0 else. Then we know that $\Pr[Y_{ij} = 1] = \frac{f}{c}$, and for a given i, the Y_{ij}'s are independent. Let $Y_i = \sum_{k=1}^{8} \sum_{j \in \mathcal{H}(k)} Y_{ij}$ (recall that $\mathcal{H}(k), 1 \leq k \leq 8$, are the cells adjacent to $\mathcal{H}$). Since the node-locations, and hence cell-populations, conform to the high probability event of Lemma 27, therefore: $E[Y_i] \leq 8 \left(\frac{6fna(n)}{5c} \right) = \frac{48}{5} \left(\frac{250f \max\{\log n, c\}}{cp_{rnd}} \right) = \frac{2400f \max\{\log n, c\}}{cp_{rnd}}$. Setting $(1 + \beta)E[Y_i] = 2400e^2 \max\{\log n, c\}$, observing from (4.13) that $\beta \geq \frac{e^2 cp_{rnd}}{f} - 1 > 0$ and applying the Chernoff bound from Lemma 51:

$$\Pr[|\mathcal{W}_i| \geq 2400e^2 \max\{\log n, c\}] = \Pr[Y_i \geq 2400e^2 \max\{\log n, c\}]$$

$$\leq \left(\frac{e^\beta}{(1+\beta)^{(1+\beta)}} \right)^{E[Y_i]}$$

$$< \left(\frac{e}{1+\beta} \right)^{(1+\beta)E[Y_i]}$$

$$\leq \left(\frac{fe}{e^2 cp_{rnd}} \right)^{2400e^2 \max\{\log n, c\}} \qquad (4.24)$$

$$= \left(\frac{f}{ecp_{rnd}} \right)^{2400e^2 \max\{\log n, c\}}$$

$$\leq \left(\frac{1}{e} \right)^{2400e^2 \max\{\log n, c\}} \qquad \left(\because \frac{f}{cp_{rnd}} \leq 1 \right)$$

$$= \exp(-2400e^2 \max\{\log n, c\}) \leq \frac{1}{n^{2400e^2}}$$

Denote by $\mathcal{E}_{i,\mathcal{H}}$ the event that, for given i and $\mathcal{H}$: $\exists \mathcal{B} \subseteq \mathcal{W}_i$ such that $|\mathcal{B}| = \lceil \frac{fna(n)}{4c} \rceil$ and $|\mathcal{C}_\mathcal{H}(\mathcal{B})| < \lceil \frac{3c}{8} \rceil$. Let $p_{ub}(x)$ be an upper-bound on $\Pr \left[\mathcal{E}_{i,\mathcal{H}} \middle| |\mathcal{W}_i| = x, c' \geq \frac{3c}{4} \right]$. Note that, having conditioned on (and hence fixed) the node-locations, $|\mathcal{W}_i|$ is independent of whether $c' \geq \frac{3c}{4}$ or not.

64

If $p_{ub}(x)$ is a non-decreasing function of x, then the following holds:

$$\Pr\left[\mathcal{E}_{i,\mathcal{H}}\Big|c' \geq \frac{3c}{4}\right]$$

$$= \Pr\left[|\mathcal{W}_i| \leq b\Big|c' \geq \frac{3c}{4}\right]\Pr\left[\mathcal{E}_{i,\mathcal{H}}\Big||\mathcal{W}_i| \leq b, c' \geq \frac{3c}{4}\right]$$

$$+ \Pr\left[|\mathcal{W}_i| > b\Big|c' \geq \frac{3c}{4}\right]\Pr\left[\mathcal{E}_{i,\mathcal{H}}\Big||\mathcal{W}_i| > b, c' \geq \frac{3c}{4}\right]$$

$$\leq \Pr\left[|\mathcal{W}_i| \leq b\right]\Pr\left[\mathcal{E}_{i,\mathcal{H}}\Big||\mathcal{W}_i| \leq b, c' \geq \frac{3c}{4}\right] + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$= \sum_{x \leq b}\Pr\left[|\mathcal{W}_i| = x\right]\Pr\left[\mathcal{E}_{i,\mathcal{H}}\Big||\mathcal{W}_i| = x, c' \geq \frac{3c}{4}\right] + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$\leq \sum_{x \leq b}\Pr\left[|\mathcal{W}_i| = x\right]p_{ub}(x) + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$\leq \sum_{x \leq b}\Pr\left[|\mathcal{W}_i| = x\right]p_{ub}(b) + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$= p_{ub}(b)\sum_{x \leq b}\Pr\left[|\mathcal{W}_i| = x\right] + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$= p_{ub}(b)\Pr\left[|\mathcal{W}_i| \leq b\right] + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$\leq p_{ub}(b) + \Pr\left[|\mathcal{W}_i| > b\right]$$

$$(4.25)$$

We now find such an upper-bound $p_{ub}(x)$ that is a non-decreasing function of x:

Note that we only need to explicitly consider $x \geq \lceil\frac{fna(n)}{4c}\rceil$, else there exist no subsets $\mathcal{B} \subseteq \mathcal{W}_i$ satisfying $|\mathcal{B}| = \lceil\frac{fna(n)}{4c}\rceil$; thus the event $\mathcal{E}_{i,\mathcal{H}}$ cannot occur, and trivially: $p_{ub}(x) = 0$ for $0 \leq x < \lceil\frac{fna(n)}{4c}\rceil$.

If $|\mathcal{W}_i| = x \geq \lceil\frac{fna(n)}{4c}\rceil$, then from Lemma 62, the number of subsets of $\mathcal{W}_i$ of cardinality $m = \lceil\frac{fna(n)}{4c}\rceil$ is given by: $\binom{x}{m} \leq \left(\frac{xe}{m}\right)^m$.

Consider a subset $\mathcal{B} \subseteq \mathcal{W}_i$ of specified cardinality $m = \lceil\frac{fna(n)}{4c}\rceil$. Denote by X_j the indicator variable which is 1 if channel j is not a member of $\mathcal{C}_{\mathcal{H}}(\mathcal{B})$ and 0 else.

Recall that each node in $\mathcal{B}$ has one channel known to be i, but the remaining $f - 1$ channels assigned to it are an i.i.d. chosen subset from the remaining $c-1$ available channels. Thus:

$$\Pr[x \in \mathcal{W}_j(j \neq i)|x \in \mathcal{W}_i] = \frac{f-1}{c-1} \geq \frac{f-1}{c} = \frac{f}{c}\left(1 - \frac{1}{f}\right) \geq \frac{99f}{100c} \quad (\because f \geq 100) \quad (4.26)$$

From (4.26), $\Pr[X_j = 1] = (1 - \frac{f-1}{c-1})^{|\mathcal{B}|} \leq (1 - \frac{99f}{100c})^{\lceil\frac{fna(n)}{4c}\rceil} \leq e^{-\frac{99f}{100c}\lceil\frac{fna(n)}{4c}\rceil}$ (applying

Fact 2). Furthermore, for a given $\mathcal{B}$, the X_j's are negatively correlated.

Let $X = \sum_{\substack{j \text{ proper in } \mathcal{H} \\ j \neq i}} X_j$. Then $E[X] \leq c'e^{-\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil}$. Setting $(1+\beta)E[X] = \frac{c'}{2}$, one

can see that $\beta = \frac{c'}{2E[X]} - 1 \geq \frac{c'}{2c'e^{-\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil}} - 1 \geq \frac{e^{\frac{99f^2na(n)}{400c^2}}}{2} - 1 \geq \frac{e^{\frac{495}{16}}}{2} - 1 > 0$ (recall that

$na(n) = \frac{250\max\{\log n,c\}}{p_{rnd}} \geq \frac{250c\max\{\log n,c\}}{2f^2} \geq \frac{125c^2}{f^2}$, from Lemma 14). Hence, we can apply

the Chernoff bound from Lemma 51 to obtain that:

$$\Pr[X \geq \frac{c'}{2}] \leq \left(\frac{e^\beta}{(1+\beta)^{(1+\beta)}}\right)^{E[X]} < \left(\frac{e}{(1+\beta)}\right)^{(1+\beta)E[X]}$$

$$= \left(\frac{2eE[X]}{c'}\right)^{\frac{c'}{2}} \leq \left(\frac{2ec'\exp(-\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil)}{c'}\right)^{\frac{c'}{2}}$$

$$= \left(2e\exp(-\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil)\right)^{\frac{c'}{2}}$$

$$= \left(\exp(-\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil + (1 + \ln 2))\right)^{\frac{c'}{2}}$$

$$\leq \left(\exp\left(-\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil + (1 + \ln 2)\right)\right)^{\frac{3c}{8}} \quad \left(\because -\frac{99f}{100c}\lceil \frac{fna(n)}{4c}\rceil + (1+\ln 2) < 0 \text{ and } c' \geq \frac{3c}{4}\right)$$

$$= \exp\left(-\frac{297f}{800}\lceil \frac{fna(n)}{4c}\rceil + \frac{3c(1 + \ln 2)}{8}\right)$$

$$< \exp\left(-\frac{297f}{800}\lceil \frac{fna(n)}{4c}\rceil + \frac{4f}{125}\lceil \frac{fna(n)}{4c}\rceil\right)$$

$$\left(\because na(n) = \frac{250\max\{\log n, c\}}{p_{rnd}} \geq \frac{250c\max\{\log n, c\}}{2f^2}, \therefore \frac{3c(1 + \log 2)}{8} < c \leq \frac{4f}{125}\lceil \frac{fna(n)}{4c}\rceil\right)$$

$$\leq \exp\left(-\frac{265f}{800}\lceil \frac{fna(n)}{4c}\rceil\right)$$

$$(4.27)$$

Due to integrality of X, $X < \frac{c'}{2} \implies X \leq \lfloor \frac{c'}{2}\rfloor \implies |\mathcal{C}_\mathcal{H}(\mathcal{B})| \geq \lceil \frac{c'}{2}\rceil \geq \lceil \frac{3c}{8}\rceil$.

Taking the union bound over all possible subsets $\mathcal{B}$, we obtain that the probability it happens for any such subset $\mathcal{B}$ is at most $\left(\frac{xe}{m}\right)^m \exp(-\frac{265f}{800}\lceil \frac{fna(n)}{4c}\rceil)$ which is an increasing function of x (recall that $m = \lceil \frac{fna(n)}{4c}\rceil$). Thus we obtain: $p_{ub}(x) = \left(\frac{xe}{m}\right)^m \exp(-\frac{265f}{800}\lceil \frac{fna(n)}{4c}\rceil)$ for $x \geq \lceil \frac{fna(n)}{4c}\rceil$. Resultantly, $p_{ub}(x)$ is an increasing function of x.

For $b = 2400e^2 \max\{\log n, c\}$:

$$p_{ub}(b) = p_{ub}(2400e^2 \max\{\log n, c\})$$

$$= \left(\frac{2400e^3 \max\{\log n, c\}}{\lceil \frac{fna(n)}{4c}\rceil}\right)^{\lceil \frac{fna(n)}{4c}\rceil} \exp\left(-\frac{265f}{800}\lceil\frac{fna(n)}{4c}\rceil\right)$$

$$\leq \left(\frac{2400e^3 \max\{\log n, c\}}{\frac{fna(n)}{4c}}\right)^{\lceil \frac{fna(n)}{4c}\rceil} \exp\left(-\frac{265f}{800}\lceil\frac{fna(n)}{4c}\rceil\right)$$

$$\leq \left(\frac{9600e^3 cp_{rnd}}{250f}\right)^{\lceil \frac{fna(n)}{4c}\rceil} \exp\left(-\frac{265f}{800}\lceil\frac{fna(n)}{4c}\rceil\right)$$

$$\leq \exp\left((3+\log\frac{960}{25}+\log\frac{cp_{rnd}}{f})\lceil\frac{fna(n)}{4c}\rceil\right) \exp\left(-\frac{265f}{800}\lceil\frac{fna(n)}{4c}\rceil\right)$$

$$< \exp\left((3+\log 40 + \log 2f)\lceil\frac{fna(n)}{4c}\rceil\right) \exp\left(-\frac{265f}{800}\lceil\frac{fna(n)}{4c}\rceil\right) \quad \text{(using Lemma 14)}$$

(4.28)

Note that:

$$\forall\, f \geq 100 : f \geq 8(3 + \log 40 + \log 2f) \tag{4.29}$$

Therefore:

$$p_{ub}(b) \leq \exp\left(\frac{f}{8}\lceil\frac{fna(n)}{4c}\rceil\right) \exp\left(-\frac{265f}{800}\lceil\frac{fna(n)}{4c}\rceil\right)$$

$$= \exp\left(-\frac{165f}{800}\lceil\frac{fna(n)}{4c}\rceil\right) < \exp\left(-\frac{f}{5}\lceil\frac{fna(n)}{4c}\rceil\right)$$

$$\leq \exp\left(-\frac{f^2na(n)}{20c}\right) \leq \exp\left(-\frac{125\log n}{20}\right) < \frac{1}{n^6}$$

(4.30)

(from Lemma 14 and our choice of $a(n)$)

From (4.24), (4.25), and (4.30): $\Pr[\mathcal{E}_{i,\mathcal{H}}|c' \geq \frac{3c}{4}] \leq p_{ub}(b) + \Pr[|\mathcal{W}_i| \geq b] \leq \frac{1}{n^6} + \frac{1}{n^{2400e^2}} < \frac{1}{n^5}$.

Since there are $c = O(\log n)$ channels i to consider, we take a union bound over them to obtain that:

$$\Pr[\mathcal{E}_{i,\mathcal{H}} \text{ for any } i \text{ in } \mathcal{H}|c' \geq \frac{3c}{4}] \leq c\Pr[\mathcal{E}_{i,\mathcal{H}} \text{ for a given } i \text{ in } \mathcal{H}|c' \geq \frac{3c}{4}])$$

Thus:

$$\Pr[\mathcal{E}_{i,\mathcal{H}} \text{ for any } i \text{ in } \mathcal{H}\ ] \leq \Pr[c' < \frac{3c}{4}] + Pr[c' \geq \frac{3c}{4}](c\Pr[\mathcal{E}_{i,\mathcal{H}} \text{ for a given } i \text{ in } \mathcal{H}|c' \geq \frac{3c}{4}])$$

$$\leq \Pr[c' < \frac{3c}{4}] + c\Pr[\mathcal{E}_{i,\mathcal{H}} \text{ for a given } i \text{ in } \mathcal{H}|c' \geq \frac{3c}{4}] \leq \frac{1}{n^2} + \frac{c}{n^5}$$

We take another union bound over all $\frac{1}{a(n)} = \frac{p_{rnd}n}{250\max\{\log n, c\}} < \frac{n}{c}$ cells $\mathcal{H}$ to obtain that the probability this occurs in any cell is at most $\frac{1}{cn} + \frac{1}{n^4}$.

Finally, recall that we conditioned our proof on the node-locations conforming to the high-probability event of Lemma 27. The probability that this event does not occur is at most $\frac{50\log n}{n}$ (as proved in Lemma 27), and we can obtain a bound by assuming that whenever that event fails to hold, the event in the statement of this lemma fails to hold.

This completes the proof that $\mathcal{C}(\mathcal{B}) \geq c' - \lfloor\frac{c'}{2}\rfloor \geq \lceil\frac{c'}{2}\rceil \geq \lceil\frac{3c}{8}\rceil$ for all specified subsets $\mathcal{B}$ of interest, for all channels i, and in all cells $\mathcal{H}$ with probability at least $1 - \frac{1}{cn} - \frac{1}{n^4} - \frac{50\log n}{n} > 1 - \frac{2}{n} - \frac{50\log n}{n}$. $\qquad\square$

4.7.1 Routing and Channel Assignment

There are two inter-related aspects of the routing procedure: determining the sequence of cells a route should traverse, and finding a feasible sequence of nodes/links along that sequence of cells which provides an end-to-end route from source to destination, while avoiding bottleneck formation.

We begin by addressing the issue of finding a feasible sequence of nodes/links that can provide an end-to-end route from source to destination, given a sequence of cells to traverse. We introduce routing structures that can facilitate this. We then show that if the number of cells traversed is at least a certain minimum number, then an end-to-end feasible route can be found, and describe a method of choosing the cell-sequence for each route. Thereafter we address the issue of constructing the routing structures in a manner that ensures load-balance.

Partial Backbones The routing strategy is based on constructing source and destination routing structures, in a manner similar to the backbones used to prove the sufficient condition for connectivity. However, *instead of constructing a full backbone for each node covering each cell of the network, only a partial backbone is constructed for each node x.*

The partial-backbone of a node x is denoted by $\mathcal{B}_p(x)$.

$\mathcal{B}_p(x)$ comprises a source segment $\mathcal{S}_b(x)$ for the flow for which x is the source. It also comprises a collection $\mathcal{D}_b(x)$ of destination segments $\mathcal{D}_b^{(i)}(x)$ for each flow i for which x is the destination. $\mathcal{S}_b(x)$ expands outwards from x to cover the sequence of cells on the route from x to its destination in that very order. Thus, there is a path comprising nodes and links in $\mathcal{S}_b(x)$ from x to any node $q_x \in \mathcal{S}_b(x)$ that follows the exact sequence of cells traversed by the route of x's flow, up to q_x's cell. Each $\mathcal{D}_b^{(i)}$ expands outwards from x to cover the cells on the route (in reverse order) from the source of flow i to x. Thus, there is a path from x to any node $q_x \in \mathcal{D}_b^{(i)}(x)$ that follows the reverse sequence of cells traversed by the route of flow i up to q_x's cell (correspondingly, the path from $q_x \in \mathcal{D}_b^{(i)}(x)$ to x follows the sequence of cells traversed by flow i's route along that stretch).

Note that each segment is a collection of nodes (V) and links/edges (E) between some of these nodes. Thus $\mathcal{S}_b(x) = (V(\mathcal{S}_b(x)), E(\mathcal{S}_b(x)))$, and $\mathcal{D}_b^{(i)}(x) = (V(\mathcal{D}_b^{(i)}(x)), E(\mathcal{D}_b^{(i)}(x)))$. Since we are concerned with load-balance, each link also has an assigned channel of operation (from amongst all feasible channels for that link).

Also note that some of the segments above may traverse common cells. In particular, x's cell is common to all segments. x is a default member of its own backbone, and all backbone segments. If two or more segments have a common cell other than x's cell, it is acceptable for each segment to have a different backbone node in that cell (and correspondingly different incoming/outgoing backbone links), if needed. Nodes/links may also be common to the segments if it is feasible while ensuring that each segment traverses the stipulated sequence of cells.

The initial part of the route of a flow i with source x and destination y is along the links of the source backbone segment $\mathcal{S}_b(x)$. As it approaches the destination, it then attempts to find a transition point and move onto the destination backbone segment $\mathcal{D}_b^{(i)}(y)$ (Fig. 4.2).

In light of the preceding lemmas, is easy to see that it is indeed always feasible to construct each segment of $\mathcal{B}_p(x)$ for all nodes x: Consider a node in some cell of the network which is the current terminus of the backbone-segment under construction. It needs to find a node in the next cell to be filled such that it can communicate with that node. The node can switch on f channels. From Lemma 32, at least $f - \lfloor \frac{f}{4} \rfloor$ of these f channels are proper
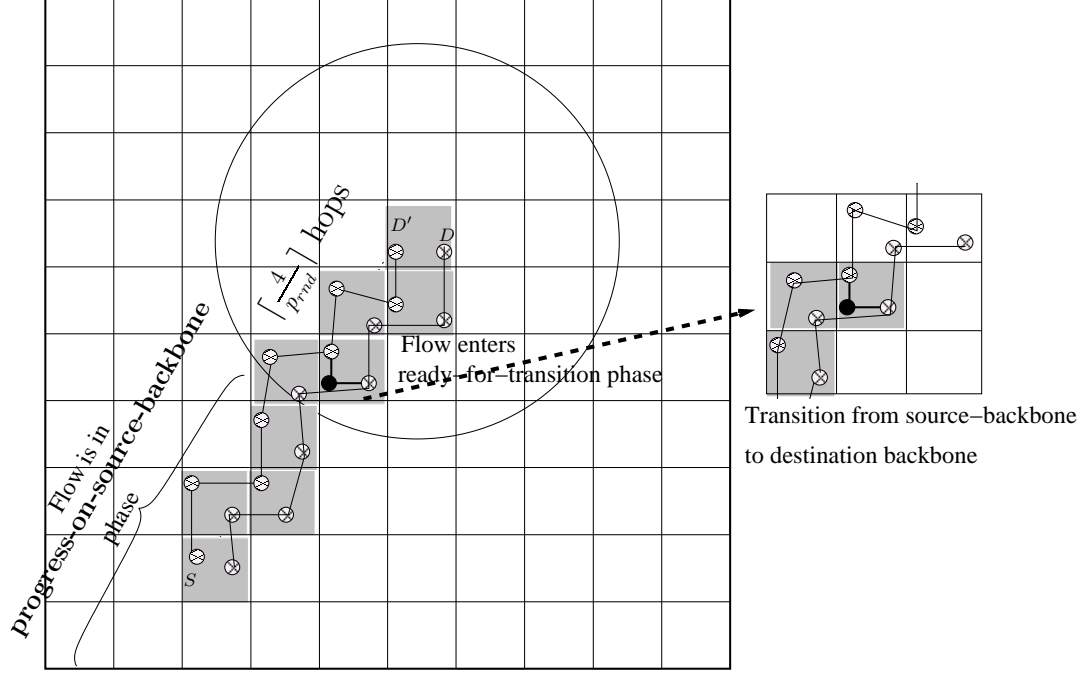
Figure 4.2: Illustration of routing along backbones

in the next cell, and therefore there are at least M_u nodes in that cell capable of switching on each of these channels w.h.p. In light of this it is always possible to expand the segment further.

However, our goal is more than just connectivity, and the backbone segments must be constructed in a manner that avoids bottleneck formation. We will later describe a backbone construction procedure that ensures load-balance. First we prove that, given any set of feasible backbones, it is possible to find an end-to-end feasible route along the backbone segments from the flow's source to its destination.

Lemma 34. *Suppose a flow i has source x and destination y. As described previously, the flow's packets are initially sent on segment $\mathcal{S}_b(x)$ of $\mathcal{B}_p(x)$ and eventually need to transition onto segment $\mathcal{D}_b^{(i)}(y)$ of $\mathcal{B}_p(y)$ (to reach y). After having traversed $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells[5] (hops) while seeking a transition opportunity, the flow will have found an opportunity to make this transition w.h.p. If the routes of each of the n flows get to traverse at least $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells (note that each individual flow's route needs to traverse at least so many distinct cells; two different flows may share cells on their respective routes), then all n flows are able to transition w.h.p.*

---

[5]The cells must be chosen in a manner independent of channel presence in the cells.

Proof. Consider a flow traversing a sequence of cells $\mathcal{H}_1, \mathcal{H}_2, \ldots$. If the representative of $\mathcal{S}_b(x)$ (let us call it q_x) in $\mathcal{H}_j$ can communicate (directly or indirectly) with the representative of $\mathcal{D}_b^{(i)}(y)$ (let us call it q_y) in $\mathcal{H}_j$, it is possible to transition from $\mathcal{S}_b(x)$ to $\mathcal{D}_b^{(i)}(y)$. If q_x and q_y can operate on some common channel, this is trivially possible. If q_x and q_y do not operate on a common channel, we consider the probability that the two can communicate via a third node from amongst the *transition facilitators* in $\mathcal{H}_j$, i.e. there exists a transition facilitator z such that z shares at least one channel with q_x and one channel with q_y. In Section 4.4.2, we showed that if q_x and q_y are incapable of direct communication, then they can communicate through a given z with probability $p_z \geq \frac{p_{rnd}^2}{40}$. Given our choice of cell area $a(n)$, and conditioned on the fact that each cell has at least $\frac{200\max\{\log n, c\}}{p_{rnd}}$ nodes (Lemma 27), of which $\frac{180\max\{\log n, c\}}{p_{rnd}}$ are deemed *backbone candidates* and the rest are *transition facilitators*, there are at least $\frac{20\max\{\log n, c\}}{p_{rnd}} \geq \frac{20\log n}{p_{rnd}}$ possibilities for z within that cell (since these cells are *intermediate* cells, i.e., do not include the cells in which x and y lie respectively, q_x and q_y themselves must be backbone candidates). All the possible z nodes have i.i.d. channel assignments. Thus, the probability that q_x and q_y cannot communicate through any z in the cell is at most $(1 - p_z)^{\frac{20\log n}{p_{rnd}}}$, and the probability they communicate through some z is $p_{xy} \geq 1 - (1 - p_z)^{\frac{20\log n}{p_{rnd}}}$.

Hence, the probability that this happens in none of the $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells is at most $(1 - p_{xy})^{\lceil \frac{4}{p_{rnd}} \rceil} \leq (1 - p_z)^{\frac{80\log n}{p_{rnd}^2}} \leq \left(1 - \frac{p_{rnd}^2}{40}\right)^{\frac{80\log n}{p_{rnd}^2}} \leq e^{-\frac{80\log n}{40}} \leq \frac{1}{n^2}$ (applying Fact 2). Applying the union bound over all n flows, the probability that all flows are able to transition is at least $1 - \frac{1}{n}$. $\qquad\square$

Therefore, we would like each route to traverse at least $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells (hops) to be able to find a transition point from the source-backbone to the destination backbone.

If the straight-line $SD'D$ path for a flow (Fig. 4.3) comprises $h \geq \lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells, it suffices to use this route. If S and D' (hence also D) lie close to each other, the hop-length of the straight line cell-to-cell path can be much smaller. In this case, a *detour* path $SPD'D$ is chosen (Fig. 4.4) in a manner similar to the previously described constructions, by choosing a point P on the circumference of a circle of radius $\frac{4}{p_{rnd}}r(n)$ centered at S. Since $r(n) = \sqrt{8a(n)}$, it is easy to see that the SP segment will traverse at least $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells.
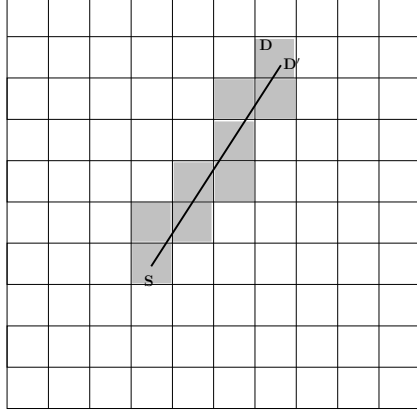
Figure 4.3: Routing along a straight line   Figure 4.4: Illustration of detour routing
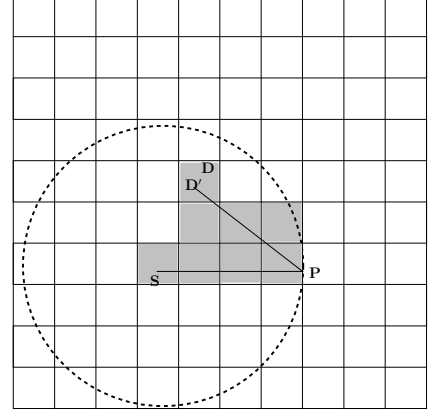
The need to perform *detour* routing for some source-destination pairs does not have any substantial effect on the relaying load on a cell.

Lemma 35. *If the number of flow-routes traversing in any cell is x when all flows use straight-line routing, it is at most $x + O(\frac{nr^2(n)}{p_{rnd}^2}) \implies x + O(\log^4 n)$ w.h.p., when detour routing is used for some of the flows as previously described.*

Proof. Recall that $c = O(\log n)$. Since the detour occurs only up to a circle of radius $\frac{4}{p_{rnd}}r(n)$, the extra flow-routes that may pass through a cell (compared to straight-line routing) are only those whose sources lie within a distance $\frac{4}{p_{rnd}}r(n)$ from some point in this cell. All such possible sources fall within a circle of radius $(1 + \frac{4}{p_{rnd}})r(n)$, and hence area $a_c(n) = \Theta(\frac{r^2(n)}{p_{rnd}^2})$. Applying Lemma 60 to the set of n node locations (with a suitable choice of $\alpha(n) \geq 1$), with high probability, any circle of this radius will have $O(na_c(n))$ nodes, and hence $O(na_c(n))$ sources. Hence, the number of extra flows that traverse the cell due to detour routing is $O(na_c(n))$, and each detour-routed flow's route can traverse a cell at most twice along the SPD' stretch. Note that the possible additional last hop for each flow is already accounted for in x. Thus, the total number of flow-routes (counting repeat traversals separately) $x + O(\frac{nr^2(n)}{p_{rnd}^2})$. Since $nr^2(n) = O(\frac{\log n}{p_{rnd}})$, and $p_{rnd} \geq \frac{f}{c}$, the total number of flow-routes is $x + O(\frac{c^3 \log n}{f^3}) \implies x + O(\log^4 n)$ w.h.p. $\square$

Flow Transition Strategy   From Lemma 34, we know that if each flow is able to inspect $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells, a transition opportunity will be found by all flows w.h.p. In light of this, we use a procedure in which there are two phases associated with the route

of a flow. A non-detour-routed flow is initially in a *progress-on-source-backbone* phase, during which its packets are sent along the links of the source backbone till there are only $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate cells left to the destination. At this point, it enters a *ready-for-transition* phase, and seeks a transition to the destination backbone along the remaining hops.[6] Once it has been able to make the transition onto the destination backbone, it proceeds towards the destination on that backbone along the remaining part of the route, and is thus guaranteed to reach the destination.

A detour-routed flow is always in *ready-for-transition* phase.

Lemma 36. *The number of flow-routes traversing any cell in* ready-for-transition *phase (counting repeat traversals separately) is $O(\log^4 n)$ w.h.p.*

Proof. First let us account for the SD' stretch of each flow's route, without considering the possible additional last hop. We account for it explicitly later in this proof.

In our construction, a non-detour routed flow enters the *ready-for-transition* phase only when it is $\lceil \frac{4}{p_{rnd}} \rceil$ distinct intermediate hops away from its destination. All such flows must have their pseudo-destinations within a circle of radius $\Theta(\frac{1}{p_{rnd}}r(n))$ centered in the cell. The number of pseudo-destinations that lie within a circle of radius $\Theta(\frac{1}{p_{rnd}}r(n))$ from the cell is $\Theta(\frac{nr^2(n)}{p_{rnd}^2}) \implies O(\frac{c^3}{f^3}\log n)$ w.h.p., (by observing that $p_{rnd} \geq \frac{f}{c}$, and using suitable choice of $\alpha(n)$ in Lemma 60). Also $c = O(\log n)$. Hence there are $O(\log^4 n)$ non-detour-routed flows in *ready-for-transition* phase traversing the cell w.h.p.

A detour-routed flow is always in *ready-for-transition* phase. By Lemma 35, there are $O(\log^4 n)$ such flows traversing any cell. Each such flow can only traverse a cell twice along the SD' (more precisely SPD') stretch. This yields $O(\log^4 n)$ detour-routed flows (including repeat traversals).

The cell may also be traversed by some of the above flows (both non-detour-routed and detour-routed) on their additional last hop. From Lemma 29, the pseudo-destinations of such flows must lie in the same cell or one of the 8 adjacent cells. Applying Lemma 59 to the set of n pseudo-destinations, the total number of pseudo-destinations lying in these 9 cells is $O(na(n))$ w.h.p. Thus, the number of flows entering the cell on their additional last hop is $O(na(n)) \implies O(\log^2 n)$ w.h.p.

---

[6]This also implies that it would suffice to construct each destination backbone segment $\mathcal{D}_b^{(i)}(x)$ for a node x only upto this distance outwards from x.
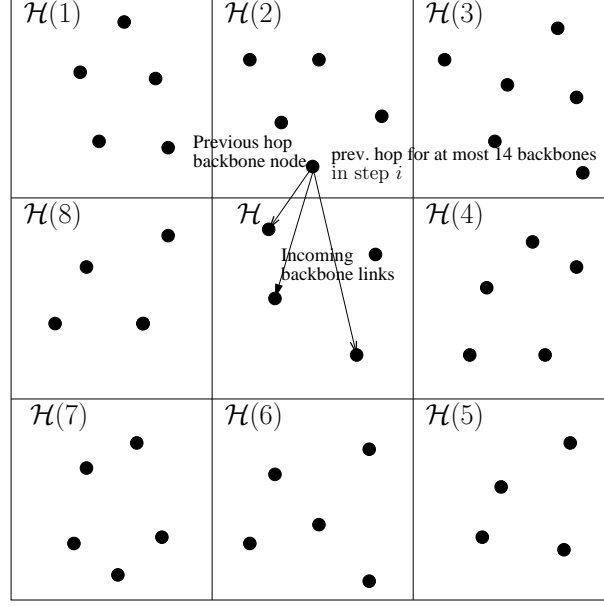
Figure 4.5: Cell $\mathcal{H}$ and neighboring cells during backbone construction

Hence, the number of flow-routes in ready-for-transition phase in any cell is $O(\log^4 n)$ w.h.p. □

Backbone Construction We now describe the procedure for constructing the backbone $\mathcal{B}_p(x)$ of x.

Given a cell $\mathcal{H}$, the 8 cells adjacent to cell $\mathcal{H}$ are denoted as $\mathcal{H}(j), 1 \le j \le 8$ (Fig. 4.5). $\mathcal{B}_p(x)$ is constructed as follows:

x is by default a member of $\mathcal{B}_p(x)$. As described earlier, $\mathcal{B}_p(x)$ has a source-segment $\mathcal{S}_b(x)$ and a collection of destination segments $\mathcal{D}_b^{(i)}(x)$ for each flow for which x is a destination.

Recall that $\mathcal{S}_b(x)$ comprises the SD' route from x to its destination, and may also have an additional last hop to D if needed. However, from Lemma 29, the only such last hop routes that may enter a cell correspond to pseudo-destinations in the 8 adjacent cells. Applying Lemma 59 to the set of pseudo-destinations, they are only $O(na(n))$ such pseudo-destinations, and hence only $O(na(n))$ such last-hop flows entering the cell. These can be accounted for separately. Therefore, we first consider the construction of the SD' part of $\mathcal{S}_b(x)$ for each node x.

Construction of $\mathcal{S}_b(x)$ Recall that we are only constructing the SD' part and not considering the possible additional last hop at this stage.

This has two sub-stages. In the first sub-stage, we construct backbones for source nodes whose flow does not require a detour. In the second sub-stage we construct backbones for source nodes whose flow requires a detour.

Straight-line backbones:

For each source of a non-detour-routed flow, the SD' segment of the route comprises the cells intersected by the straight-line SD'. One can define an ordering on these cells that reflects the order in which each cell is encountered when moving from S to D' along the straight-line. The backbone-segment $\mathcal{S}_b(x)$ is expanded into new cells in the same order.

This step proceeds in a synchronized hop-by-hop manner for all non-detour-routed flows (each of which has a unique source x).

Any cell of $\mathcal{S}_b(x)$ in which there is already a node assigned to $\mathcal{S}_b(x)$ is called a *filled* cell. Thus, initially x's cell is filled. We consider the cell in $\mathcal{S}_b(x)$ that is entered next by the flow's straight-line route. We consider all nodes in that cell that can operate on one or more common channel with x. This provides a number of alternative channels on which the flow's backbone can enter that cell.

Let h_{max} be the maximum hop-length of any non-detour-routed SD' route. Then, $h_{max} = O(\frac{1}{\sqrt{a(n)}})$ and the procedure has h_{max} steps. In step k, for each source node x whose flow has k or more hops, $\mathcal{S}_b(x)$ expands into the cell entered by x's flow on the k-th hop.

Each cell $\mathcal{H}$ performs the procedure we will now describe.

Lemma 37. *If $f \geq 100$, then it is possible to devise a backbone construction procedure, such that, after step h_{max} of the backbone construction procedure for the SD' part of $\mathcal{S}_b(x)$ (for sources x whose flows are not detour-routed), each cell has $O(\frac{n\sqrt{a(n)}}{c})$ incoming backbone links on a single channel, and each node appears on $O(\frac{n\sqrt{a(n)}}{c})$ (source) backbones, w.h.p.*

Proof. We describe such a backbone construction procedure and prove its load-balance characteristics by induction.

We remark at the outset that the proof is conditioned on the occurrence of the high probability events in Lemma 27, Lemma 28, Lemma 32, and Lemma 33.

Recall that we are expanding backbones to cover cells in $\mathcal{S}_b(x)$.

At each step of the construction, we first have a channel-allocation phase, followed by a node-allocation phase. We prove that after step k of the backbone construction procedure,

75

the following two invariants hold for *all* cells of the network:

- *Invariant 1:* Each node is assigned at most 14 new incoming backbone links during step k. Thus after step k, it appears in a total of $O(14k) \implies O(k)$ backbones.

- *Invariant 2:* No more than $\lfloor \frac{5na(n)}{c} \rfloor$ new backbone links enter the cell on a single channel during step k. Thus, $O(\frac{kna(n)}{c})$ incoming backbones (entering the cell) are assigned (incoming links) on any single channel after step k.

If the above two Invariants hold, then it is easy to see that after h_{max} steps, cell $\mathcal{H}$ will have no more than $\frac{5h_{max}na(n)}{c} = O(\frac{n\sqrt{a(n)}}{c})$ backbone links assigned to any single channel, and no node occurs on more than $14h_{max} \implies O(\frac{1}{\sqrt{a(n)}}) \implies O(\frac{n\sqrt{a(n)}}{c})$ backbones (from (4.15)).

We prove by induction that the invariants hold, as follows:

If Invariant 1 holds at the end of step $k-1$, then Invariant 2 continues to hold after the channel-allocation phase of step k. If Invariant 2 holds after the channel-allocation phase of step k, then Invariant 1 will continue to hold after the node-allocation phase of step k, and thus both Invariants 1 and 2 will hold at the end of step k.

Base Case: Before the procedure begins, at step 0, each node is assigned to its own backbone, for which it is effectively the origin (this can also be viewed as a single backbone link incoming to this node from an imaginary super-source). Thus, after Step 0, Invariant 1 holds trivially. Invariant 2 is trivially true.

Inductive Step:

Suppose Invariants 1 and 2 held at the end of step $k-1$. Consider a particular cell $\mathcal{H}$ during step k.

Let the number of *proper* channels in $\mathcal{H}$ be c'.

From Lemma 32, $c' \geq c - \lfloor \frac{f}{4} \rfloor \geq \frac{3c}{4}$ for each cell. Each backbone $\mathcal{S}_b(x)$ that enters cell $\mathcal{H}$ in step k has a previous hop-node in one of the 8 adjacent cells. Also note that, as a consequence of Lemma 32, each previous hop node has at least $\lceil \frac{3f}{4} \rceil$ of cell $\mathcal{H}$'s *proper* channels available to it as choices for the link that will enter cell $\mathcal{H}$ (since it can operate on f channels, of which at most $\lfloor \frac{f}{4} \rfloor$ can be non-proper in cell $\mathcal{H}$).
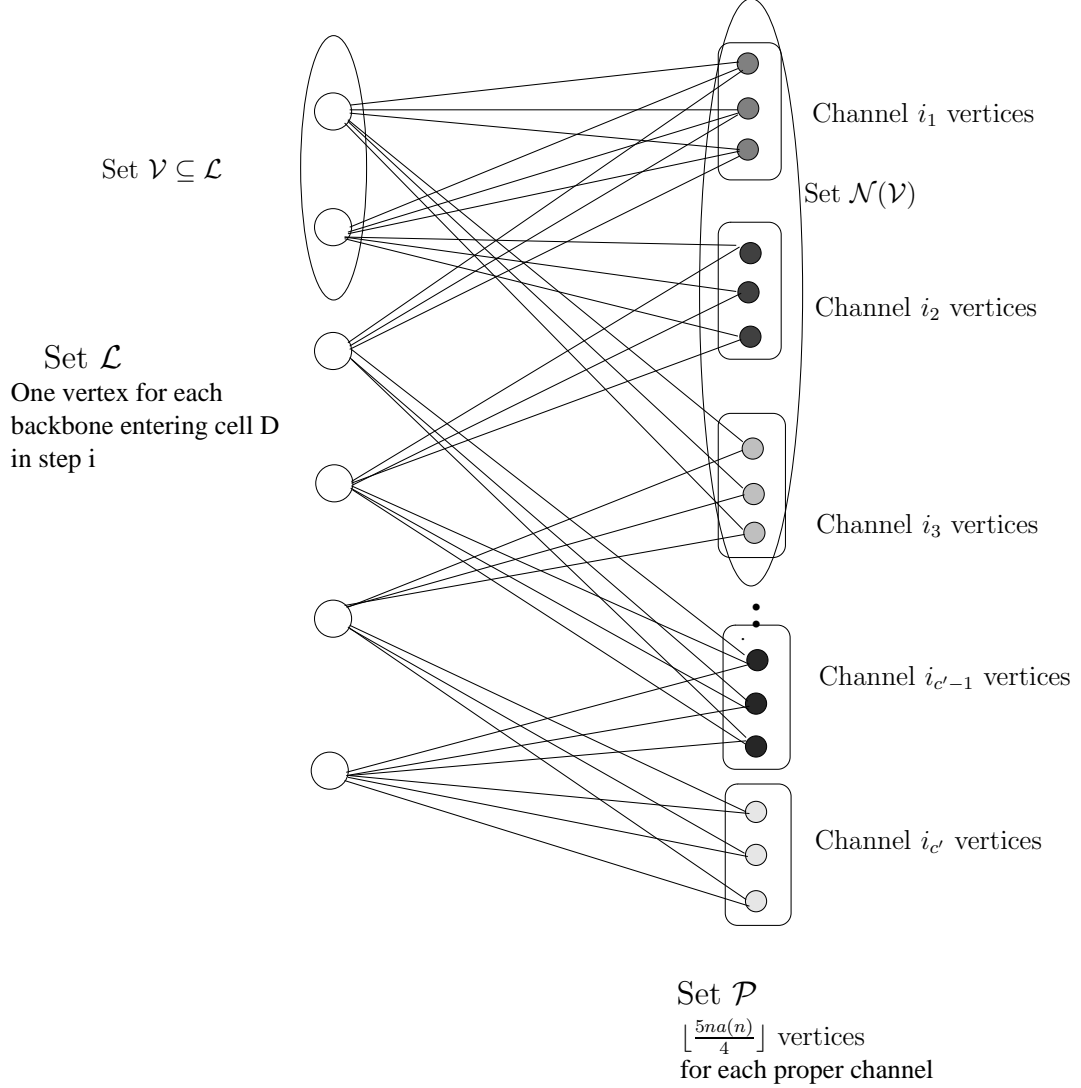
Figure 4.6: Bipartite Graph for Cell $\mathcal{H}$ in step k

Channel-Allocation Construct a bipartite graph with two sets of vertices (Fig. 4.6): one set (call it $\mathcal{L}$) has a vertex corresponding to each of the (source) backbones that enter the cell $\mathcal{H}$ in step k. From Lemma 28, it follows that $|\mathcal{L}| \leq \lfloor \frac{5na(n)}{4} \rfloor$. The other set (call it $\mathcal{P}$) has $\lfloor \frac{5na(n)}{c} \rfloor \leq \frac{5na(n)}{c}$ vertices for each proper channel i in cell $\mathcal{H}$, i.e., $|\mathcal{P}| = c' \lfloor \frac{5na(n)}{c} \rfloor$.

A backbone vertex is connected to all the vertices for the channels proper in $\mathcal{H}$ on which the previous hop node of that backbone can switch (and which are therefore valid channel choices for entering the cell $\mathcal{H}$). We show that there exists a matching that pairs each backbone vertex to a unique channel vertex, through an argument based on Hall's marriage theorem (Theorem 31). Thus, our objective is to show that for all $\mathcal{V} \subseteq \mathcal{L}$, $|\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$,

where $\mathcal{N}(\mathcal{V}) \subseteq \mathcal{P}$ is the union of the neighbor-sets of all vertices in $\mathcal{V}$.

We first note the following:

$$
\begin{aligned}
\lceil \frac{3f}{4} \rceil \lfloor \frac{5na(n)}{c} \rfloor &\geq \frac{3f}{4} \left(\frac{5na(n)}{c} - 1 \right) = \frac{15fna(n)}{4c} - \frac{3f}{4} \\
&\geq \frac{15fna(n)}{4c} - \frac{3fna(n)}{1000c} \geq \frac{29fna(n)}{8c} \quad (\because na(n) \geq 250c)
\end{aligned}
\tag{4.31}
$$

Consider the following two cases:

Case 1: $|\mathcal{V}| < \frac{29fna(n)}{8c}$

Consider any set $\mathcal{V}$ of backbone vertices such that $|\mathcal{V}| < \frac{29fna(n)}{8c}$. Then, since there are at most $\lfloor \frac{f}{4} \rfloor$ non-proper channels in a cell, every previous hop node has at least $\lceil \frac{3f}{4} \rceil \geq \frac{3f}{4}$ *proper* channel choices. For each proper channel there are $\lfloor \frac{5na(n)}{c} \rfloor \geq \frac{5na(n)}{c} - 1$ associated channel vertices. Using (4.31), we obtain that: $|\mathcal{N}(\mathcal{V})| \geq \lceil \frac{3f}{4} \rceil \lfloor \frac{5na(n)}{c} \rfloor \geq \frac{29fna(n)}{8c}$. Therefore $|\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$.

Case 2: $|\mathcal{V}| \geq \frac{29fna(n)}{8c}$

Consider sets $\mathcal{V}$ of size at least $\frac{29fna(n)}{8c}$. Intuitively, to show that $|\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$ for all such $\mathcal{V}$, we first show that if a channel overload condition occurs, resulting in $|\mathcal{N}(\mathcal{V})| < |\mathcal{V}|$ for some $\mathcal{V}$, then the overload must also manifest itself in some *channel-aligned* subset (i.e., a subset where all incoming backbones corresponding to subset vertices have some *common* proper channel i available to them). Thus, to show that no overload condition occurs, it suffices to show that no overload condition occurs in any of these *critical* channel-aligned subsets, which can be shown using Lemma 33. The argument is formalized as follows:

Let $\mathcal{V}_i$ be the set comprising all sets $\mathcal{U}_i \subseteq \mathcal{L}$, such that all backbone vertices in $\mathcal{U}_i$ have channel i associated with them (i.e., all backbone vertices in $\mathcal{U}_i$ have i available to them as a valid proper channel choice for entering $\mathcal{H}$).

Claim (a) $\forall \mathcal{U} \in \bigcup_{i \text{ proper in } \mathcal{H}} \mathcal{V}_i :$

$$
\text{If } |\mathcal{U}| \geq \lceil \frac{29fna(n)}{8c} \rceil \text{ then } |\mathcal{N}(\mathcal{U})| \geq |\mathcal{L}|
$$

Proof of Claim (a): By assumption, $\mathcal{U} \in \mathcal{V}_i$ for some i that is proper in $\mathcal{H}$. Also, since no node can be the previous hop in step k of more flows than those assigned to it in step $k-1$, and Invariant 1 held after step $k-1$, therefore no previous hop node is common to more than 14 backbone links entering $\mathcal{H}$ in step k. Let $\mathcal{A}$ be the set of distinct previous hop nodes associated with $\mathcal{U}$. If $|\mathcal{U}| \geq \lceil \frac{29fna(n)}{8c} \rceil$, then $|\mathcal{A}| \geq \frac{1}{14}|\mathcal{U}| \geq \frac{1}{14}(\frac{29fna(n)}{8c}) \geq \frac{fna(n)}{4c} + \frac{fna(n)}{112c} > \frac{fna(n)}{4c} + 1 \geq \lceil \frac{fna(n)}{4c} \rceil$ (note that $\frac{fna(n)}{c} \geq 250f \geq 500 > 112$).

Therefore, $\mathcal{A}$ contains at least one subset $\mathcal{B}$ satisfying $|\mathcal{B}| = \lceil \frac{fna(n)}{4c} \rceil$. Recognizing that all members of $\mathcal{A}$, and hence all members of $\mathcal{B}$, are capable of switching on channel i, we can invoke Lemma 33 on $\mathcal{B}$, to obtain that when $f \geq 100$: $|\mathcal{C}_{\mathcal{H}}(\mathcal{B})| \geq \lceil \frac{3c}{8} \rceil$. This yields: $\mathcal{N}(\mathcal{U}) \geq |\mathcal{C}_{\mathcal{H}}(\mathcal{B})| \lfloor \frac{5na(n)}{c} \rfloor \geq |\mathcal{C}(\mathcal{B})| \left(\frac{5na(n)}{c} - 1 \right) \geq \lceil \frac{3c}{8} \rceil \left(\frac{5na(n)}{c} - 1 \right) \geq \frac{15na(n)}{8} - \lceil \frac{3c}{8} \rceil \geq \frac{15na(n)}{8} - \frac{3}{8} \left(\frac{na(n)}{250} \right) - 1 \geq \frac{5na(n)}{4} \geq |\mathcal{L}|$.

Claim (b)   Consider a set $\mathcal{V} \subseteq \mathcal{L}$.

If $|\mathcal{N}(\mathcal{V})| < |\mathcal{V}|$ then $\exists$ channel i proper in $\mathcal{H}$, and $\mathcal{S}_i \subseteq \mathcal{V}$ such that:
$$\mathcal{S}_i \in \mathcal{V}_i \text{ and } |\mathcal{S}_i| \geq \lceil \frac{29fna(n)}{8c} \rceil \tag{4.32}$$

Proof of Claim (b): Suppose $|\mathcal{N}(\mathcal{V})| < |\mathcal{V}|$. Let us denote by $\mathcal{S}_i \subseteq \mathcal{V}$ the set of all backbone vertices in $\mathcal{V}$ that are associated with channel i (i.e., have channel i available as a valid proper channel choice for entering cell $\mathcal{H}$). Consider the bipartite sub-graph $G_{\mathcal{V}}$ induced by $\mathcal{V} \cup \mathcal{N}(\mathcal{V})$, and assign all edges unit capacity. Construct the graph $G_{\mathcal{V}}' = (\mathcal{V} \cup \mathcal{N}(\mathcal{V}) \cup \{s,t\}, E)$ where s is a source node having a unit capacity edge to all vertices $v \in \mathcal{V}$, and t is a sink node, connected to each vertex $u \in \mathcal{N}(\mathcal{V})$ via a unit capacity edge (thus, E comprises the edges in $G_{\mathcal{V}}$ and the additional edges just described).

We try to obtain a (s,t) flow g in $G_{\mathcal{V}}'$ such that all edges (s,v) are saturated. Each vertex $v \in \mathcal{V}$ sub-divides the unit of flow received from s equally amongst all edges (v,u) outgoing from it. Since each vertex has edges to vertices of at least $\lceil \frac{3f}{4} \rceil$ channels, this yields at least $\lceil \frac{3f}{4} \rceil \lfloor \frac{5na(n)}{c} \rfloor \geq \frac{3f}{4} \left(\frac{5na(n)}{c} - 1 \right) \geq \frac{29fna(n)}{8c}$ edges (see (4.31)). Thus, each $v \in \mathcal{V}$ contributes at most $\frac{8c}{29fna(n)}$ units of flow to a vertex $u \in \mathcal{N}(\mathcal{V})$, i.e., $g(v,u) \leq \frac{8c}{29fna(n)}$. Hence no vertex $u \in \mathcal{N}(\mathcal{V})$ gets more than $h(u) = \sum_{v \in \mathcal{S}_i} g(v,u) = \frac{8c|\mathcal{S}_i|}{29fna(n)}$ units of flow, where i is the channel corresponding to vertex u. Resultantly, if $|\mathcal{S}_i| \leq \lfloor \frac{29fna(n)}{8c} \rfloor$ for all channels i that are proper in cell $\mathcal{H}$, this implies that $h(u) \leq 1$, and setting $g(u,t) = h(u)$ yields the

desired (s, t) flow. Hence g is a valid flow that allows a unit of flow to pass through each vertex $v \in \mathcal{V}$. Therefore, from the Integrality Theorem (Theorem 32), we can obtain an integer-capacity flow, which yields a matching of size $|\mathcal{V}|$. Therefore, from Hall's marriage theorem (Theorem 31), $|\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$ (else a matching of size $|\mathcal{V}|$ could not have existed). This yields a contradiction. Hence, there must exist a proper channel i, and $\mathcal{S}_i \subseteq \mathcal{V}$ such that $\mathcal{S}_i \in \mathcal{V}_i$ and $|\mathcal{S}_i| > \lfloor \frac{29fna(n)}{8c} \rfloor$. Since set-cardinality must necessarily be an integer, it follows that $|\mathcal{S}_i| \geq \lceil \frac{29fna(n)}{8c} \rceil$, and (4.32) holds.

Claim (c) $\forall \mathcal{V} \subseteq \mathcal{L}$ such that $|\mathcal{V}| \geq \frac{29fna(n)}{8c} : |\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$

   Proof of Claim (c): Suppose $|\mathcal{N}(\mathcal{V})| < |\mathcal{V}|$. Then, from Claim (b), there exists a set $\mathcal{S}_i \subseteq \mathcal{V}$ such that $\mathcal{S}_i \in \mathcal{V}_i$, and $|\mathcal{S}_i| \geq \lceil \frac{29fna(n)}{8c} \rceil$. Thus $\mathcal{S}_i$ qualifies as a set to which Claim (a) applies. Invoking Claim (a) on this set $\mathcal{S}_i$, it follows that $|\mathcal{N}(\mathcal{V})| \geq |\mathcal{N}(\mathcal{S}_i)| \geq |\mathcal{L}| \geq |\mathcal{V}|$. This yields a contradiction. Thus, $|\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$.

   Taking both Case 1 and Case 2 into account, we have thus proved that $\forall \mathcal{V} \subseteq \mathcal{L} : |\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$. Therefore, from Hall's marriage theorem (Theorem 31), each backbone vertex can be matched with a unique channel vertex, and the corresponding backbone will be assigned to the channel with which this vertex is associated. Thus all backbones get assigned a channel, and (since there are $\lfloor \frac{5na(n)}{c} \rfloor$ channel vertices for each proper channel) no more than $\lfloor \frac{5na(n)}{c} \rfloor$ incoming backbone links are assigned to any single channel.

   While Hall's marriage theorem proves that such a matching exists, the matching itself can be computed using the Ford-Fulkerson method [22] on a flow network obtained from the bipartite graph by adding a source with an edge to each vertex in $\mathcal{L}$, a sink to which each vertex in $\mathcal{P}$ has an edge, and assigning unit capacity to all edges.

   Thus, Invariant 2 continues to hold after the channel-allocation phase of step k.

Node-Allocation   Having determined the channel each incoming backbone link should use to enter cell $\mathcal{H}$, we need to assign a node in cell $\mathcal{H}$ to each backbone. For this, we again construct a bipartite graph. In this graph, the first set of vertices (call it $\mathcal{F}$) comprise a vertex for each backbone link entering cell $\mathcal{H}$ in step k. The second set (call it $\mathcal{R}$) comprises 14 vertices for each *backbone candidate* node in cell $\mathcal{H}$. A vertex x in $\mathcal{F}$ has an edge with a vertex y in $\mathcal{R}$ iff the actual *backbone candidate* node associated with y is capable of switching on the channel assigned to the backbone-link associated with vertex x in the

preceding channel-allocation phase (this implies that y is indeed a valid relay choice for the backbone link corersponding to x).

Each vertex $x \in \mathcal{F}$ has degree at least $14M_u$, since it is assigned to a *proper* channel, which by definition has at least M_u representatives in cell $\mathcal{H}$, each of which has 14 associated vertices in $\mathcal{R}$. Also recall that $M_u = \lceil \frac{9fna(n)}{25c} \rceil$. Once again we seek to show that for all $\mathcal{V} \subseteq \mathcal{F}, |\mathcal{N}(\mathcal{V})| \geq |\mathcal{V}|$.

Consider any set $\mathcal{V} \in \mathcal{F}$.

Since no channel is assigned more than $\lfloor \frac{5na(n)}{c} \rfloor$ entering backbone links during the channel-allocation phase of this step, the vertices in $\mathcal{V}$ are cumulatively associated with at least $m \geq \frac{|\mathcal{V}|}{\lfloor \frac{5na(n)}{c} \rfloor}$ distinct proper channels. Since each of these channels has at least M_u *backbone candidate* nodes capable of switching on them, and any one node can only switch on up to f proper channels, this implies that the number of distinct nodes in cell $\mathcal{H}$ cumulatively associated with these $m \geq \frac{|\mathcal{V}|}{\lfloor \frac{5na(n)}{c} \rfloor}$ proper channels is at least $\frac{|\mathcal{V}|M_u}{f\lfloor \frac{5na(n)}{c} \rfloor} \geq \frac{|\mathcal{V}|\lceil \frac{9fna(n)}{25c} \rceil}{\frac{5fna(n)}{c}} \geq \frac{9|\mathcal{V}|}{125}$. Since each backbone candidate node has 14 vertices in $\mathcal{R}$, it follows that $|\mathcal{N}(\mathcal{V})| \geq 14\left(\frac{9|\mathcal{V}|}{125}\right) \geq \frac{126|\mathcal{V}|}{125} > |\mathcal{V}|$.

Then invoking Hall's Marriage Theorem again, each vertex $x \in \mathcal{F}$ can be matched with a unique vertex $y \in \mathcal{R}$, and the actual network node associated with y is deemed the backbone representative for the backbone corresponding to vertex x in cell $\mathcal{H}$ (the matching can again be computed via the Ford-Fulkerson method). Since there are at most 14 vertices associated with a node, no node is assigned more than 14 incoming backbone links in step k, and Invariant 1 continues to hold after the node-allocation phase of step k.

This proves that both Invariants 1 and 2 continue to hold after step k.

It follows that, after step h_{max} (where $h_{max} \leq \frac{2}{\sqrt{a(n)}}$), each cell $\mathcal{H}$ has $O(\frac{h_{max}na(n)}{c}) \implies O(\frac{n\sqrt{a(n)}}{c})$ entering backbone links per channel, and each node appears on $O(h_{max}) = O(\frac{1}{\sqrt{a(n)}}) \implies O(\frac{n\sqrt{a(n)}}{c})$ (from (4.15)) source backbones. $\square$

Detour backbones: We can construct the SPD' stretch of backbone segment $\mathcal{S}_b(x)$ for the detour-routed flows in any manner possible, i.e., by assigning links to any eligible node/channel (at least one eligible node is known to exist since, as a consequence of Lemma 32, each node can switch on at least $\lceil \frac{3f}{4} \rceil$ channels that are proper in the next cell).

Additional last hop: Now let us account for the possible additional last hop that some

flows may have, yielding an additional cell in $\mathcal{S}_b(x)$ (in addition to those traversed from source x to pseudo-destination). We can extend the backbones over the additional hop in any feasible manner (and as argued for the detour backbones, it is indeed feasible to do so).

Construction of $\mathcal{D}_b(x)$ Note that by our routing strategy a flow will only attempt to transition to the destination backbone when it enters *ready-for-transition* phase.

From Lemma 36, the total number of flows-routes traversing a cell in *ready-for-transition* phase is $O(\log^4 n)$ (counting possible repeat traversals), which is asymptotically dominated by $O(\frac{n\sqrt{a(n)}}{c})$.

Therefore, for each node x, and for each flow i for which x is the destination: we can construct $\mathcal{D}_b^{(i)}(x)$ by using any feasible nodes/channels (it is always feasible to construct $\mathcal{D}_b^{(i)}(x)$ as each node can switch on at least $\lceil \frac{3f}{4} \rceil$ channels that are proper in the next cell to be traversed).

4.7.2   Load Balance within a Cell

Now we show that no channel or interface bottlenecks form in the network when our described construction is used. As in Section 4.6, we use the following terminology: A flow-link is said to *enter* a cell $\mathcal{H}$ on a channel j if the flow's route includes a hop (link) (v_{i-1}, v_i), where v_{i-1} is in a cell adjacent to $\mathcal{H}$, v_i is in $\mathcal{H}$, and v_{i-1} transmits the flow's packets to v_i using channel j (this naturally implies that both v_{i-1} and v_i can operate on channel j). Similarly, a flow-link is said to leave a cell $\mathcal{H}$ on channel j if the route includes a link (v_i, v_{i+1}), where v_i is in $\mathcal{H}$, v_{i+1} is in a cell adjacent to $\mathcal{H}$, and v_i transmits the flow's packets to v_{i+1} using channel j.

Per-Channel Load

Lemma 38. *The number of flow-links that enter any cell on a given channel is $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p.*

Proof. A flow-route traversing $\mathcal{H}_1, \mathcal{H}_2, ..., \mathcal{H}_{j-1}, \mathcal{H}_j....$ may enter a cell $\mathcal{H}_j$ on a channel i under the following circumstances:

1. The flow is either in *progress-on-source-backbone* phase, or it is in the *ready-for-transition* phase, but is yet to make a transition to the destination backbone, and

i is the channel assigned to the source backbone link between the backbone nodes in $\mathcal{H}_{j-1}$ and $\mathcal{H}_j$

2. The flow has already made a transition, and i is the channel assigned to the link between the destination backbone nodes in $\mathcal{H}_{j-1}$ and $\mathcal{H}_j$

We first consider the flow-links that enter a cell in *progress-on-source-backbone* phase, i.e., they are proceeding on their respective source backbone segments. Recall that these are all non-detour-routed flows, since detour-routed flows are always in *ready-for-transition* phase. The number of such flows that enter any cell on a single channel is $O(\frac{n\sqrt{a(n)}}{c})$ (Lemma 37).

We now need to account for the fact that some of the flow-links may enter the cell in the *ready-for-transition* phase. From Lemma 36 there are $O(\log^4 n)$ flow-routes traversing any cell in *ready-for-transition* phase w.h.p. (recall that these include the detour-routed flows with their repeat traversals counted separately, and also the possible additional last $D'D$ hop for all flows). Thus, regardless of whether they are still on their source backbone, or have already made the transition to their destination backbone, the number of such entering flow-links assigned to any single channel is $O(\log^4 n)$.

Hence the number of flow-links entering on a single channel is $O(\frac{n\sqrt{a(n)}}{c}) + O(\log^4 n) \implies O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. for each cell of the network. $\qquad \square$

Lemma 39. *The number of flow-links that leave any cell on any single channel is* $O(\frac{n\sqrt{a(n)}}{c})$ *w.h.p.*

Proof. Note that the flow-links that leave the cell must then enter one of the 8 adjacent cells on that channel (as a backbone link for a flow leaves the current cell, and enters an adjacent cell). Hence, flow-links leaving the cell on a channel can be no more than 8 times the maximum number of flow-links entering a cell on any one channel, which has been established as $O(\frac{n\sqrt{a(n)}}{c})$ in Lemma 38. Therefore, the total number of flows leaving any given cell on a given channel is also $O(\frac{n\sqrt{a(n)}}{c})$ w.h.p. $\qquad \square$

Lemma 40. *The number of additional transition links scheduled on any single channel within any cell is* $O(\log^4 n)$ *w.h.p.*
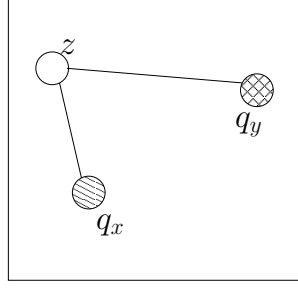
Figure 4.7: Two additional transition links for a flow lying wholly within the cell

Proof. Recall the transition strategy outlined in the proof of Lemma 34, whereby the flow locates a cell along the route where the source backbone node q_x, and destination backbone node q_y are connected through a third node z. This yields two additional links $q_x \rightarrow z$, and $z \rightarrow q_y$ that lie entirely within the cell (Fig. 4.7). Note that the number of flows performing this transition in the cell can be no more than the number of flows traversing the cell in *ready-for-transition* phase. From Lemma 36 there are $O(\log^4 n)$ such flows traversing any cell w.h.p. In the worst case, we can count 2 additional links for each such flow as being all assigned to one channel. The result follows from this observation. $\square$

Per-Node Load

Lemma 41. *The number of flow-links that are assigned to any one node in any cell is* $O(\frac{n\sqrt{a(n)}}{c})$ *w.h.p.*

Proof. A node is always assigned an outgoing flow-link for the single flow for which it is the source. A node is also assigned an incoming flow-link for each flow for which it is the destination, and from Lemma 1 there are $O(\log n)$ such flows for any node w.h.p. Besides, a node may be assigned a pair of flow-links (incoming and outgoing) for flows that are in the *ready-to-transition* phase, for which it facilitates a transition (if it is a transition facilitator) node), or on whose source or destination backbone it occurs (if it is a backbone candidate). There are $O(\log^4 n)$ such flow-links (counting repeat traversals by the same flow, additional last hop, and additional transition links separately) in a cell w.h.p. (Lemma 36 and Lemma 40). Thus, a node can only have $O(\log^4 n)$ such flow-links assigned.

We now consider the flows in *progress-on-source-backbone* phase that do not originate in the cell. Note that these must be non-detour-routed flows in their SD' stretch. These flows are on their source-backbone, and from Lemma 37, each *backbone candidate* node has

$O(\frac{n\sqrt{a(n)}}{c})$ incoming flow-links assigned. Corresponding to each such incoming link, there is an outgoing link (since the node is a relay for these flows). Thus, the total number of such assigned flow-links is $O(\frac{n\sqrt{a(n)}}{c})$.

Therefore, the number of flow-links assigned to any single node is $1 + O(\log n) + O(\log^4 n) + O(\frac{n\sqrt{a(n)}}{c}) \implies O(\frac{n\sqrt{a(n)}}{c})$. $\qquad\qquad\square$

4.7.3 Transmission Schedule

Similar to adjacent (c, f) assignment, and the sub-optimal lower bound construction of Section 4.6, we can obtain a two-level feasible transmission schedule. Since, each cell can face interference from at most a constant number γ of nearby cells, the resultant cell-interference graph (a graph with a vertex for each cell, and an edge between two vertices if the corresponding cells can interfere with each other), has a chromatic number at most $1 + \gamma$. Hence, we can come up with a global schedule having $1 + \gamma$ unit time slots in each round. In any slot, if a cell is active, then all interfering cells are inactive.

For intra-cell scheduling, we construct a conflict graph based on the nodes in the active cell, and its adjacent cells (note that the hop-sender of each flow shall lie in the active cell, and the hop-receiver shall lie in one of the adjacent cells, except for transition links, for which both lie in the active cell), as follows:

We create a separate vertex for each flow-link for which a node in the cell needs to transmit data (repeat traversals by the same flow's route or additional transition links lying wholly within the cell are counted as distinct flow-links for the purpose of scheduling; these have been accounted for while bounding the number of flow-links in a cell in previous lemmas). Since each flow-link has an assigned channel on which it operates, each vertex in the graph has an implicit associated channel. Besides, each vertex has an associated pair of nodes corresponding to the hop endpoints. Two vertices are connected by an edge if (1) they have the same associated channel, or (2) at least one of their associated nodes is the same. The scheduling problem thus reduces to obtaining a vertex-coloring of this graph. If we have a vertex coloring, then it ensures that (1) a node is never simultaneously sending/receiving for more than one flow (2) no two flow-links on the same channel are active simultaneously. Thus, the number of neighbors of a graph vertex is upper bounded by the number of flow-links requiring a transmission in the active cell on that channel, and

the number of flow-links assigned to the flow's two hop endpoints (both hop-sender and hop-receiver). It can be seen from Lemma 39, Lemma 40 and Lemma 41 that the degree of the conflict graph is $O(\frac{n\sqrt{a(n)}}{c})+O(\frac{n\sqrt{a(n)}}{c})+O(\log^4 n)+O(\frac{n\sqrt{a(n)}}{c})+O(\frac{n\sqrt{a(n)}}{c})=O(\frac{n\sqrt{a(n)}}{c})$ (note that $O(\log^4 n)\implies O(\frac{n\sqrt{a(n)}}{c})$, since we showed in (4.14) that $\frac{n\sqrt{a(n)}}{c}=\Omega(\sqrt{\frac{n}{\log n}})$). Thus the graph can be colored in $O(\frac{n\sqrt{a(n)}}{c})$ colors. Hence, the cell-slot (which can be assumed to be of unit time) is divided into $O(\frac{n\sqrt{a(n)}}{c})=O(\frac{\sqrt{\frac{n\log n}{p_{rnd}}}}{c})$ equal length subslots, and all the flow-links get a slot for transmission. This implies that each flow-link gets a $\Omega(c\sqrt{\frac{p_{rnd}}{n\log n}})$ fraction of the slot-time. Moreover, each cell gets at least one slot in $1+\gamma$ slots, where γ is a constant, and each channel has bandwidth $\frac{W}{c}$. Thus, the throughput each flow can get is $\Omega\left(\left(\frac{1}{1+\gamma}\right)\left(\frac{W}{c}\right)\left(c\sqrt{\frac{p_{rnd}}{n\log n}}\right)\right)=\Omega(W\sqrt{\frac{p_{rnd}}{n\log n}})$.

Theorem 7. *When $c=O(\log n)$ and $100\leq f\leq c$, construction CR_2 yields a per-flow throughput of $\Omega(W\sqrt{\frac{p_{rnd}}{n\log n}})$ for random (c,f) assignment.*

We now describe the construction CR^*.

Construction CR^*

- *When $f<100$:* Use construction CR_1 described in Section 4.6, which achieves a per-flow throughput of $\Omega(W\sqrt{\frac{f}{cn\log n}})$ (Theorem 6). From Lemma 14, it follows that $\frac{\sqrt{\frac{f}{cn\log n}}}{\sqrt{\frac{p_{rnd}}{n\log n}}}=\Omega(\frac{1}{\sqrt{f}})$. Thus, for $f<100$, $\frac{\sqrt{\frac{f}{cn\log n}}}{\sqrt{\frac{p_{rnd}}{n\log n}}}=\Omega(1)$.

- *When $f\geq 100$:* Use construction CR_2, which achieves a per-flow throughput of $\Omega(W\sqrt{\frac{p_{rnd}}{n\log n}})$ whenever $f\geq 100$ (Theorem 7).

This yields the following result:

Theorem 8. *When $c=O(\log n)$ and $2\leq f\leq c$, construction CR^* yields a per-flow throughput of $\Omega(W\sqrt{\frac{p_{rnd}}{n\log n}})$.*

Combining Theorem 8 with the upper bound on capacity proved in Section 4.5, we obtain the following theorem:

Theorem 9. *When $c=O(\log n)$ and $2\leq f\leq c$, the per-flow network capacity with random (c,f) assignment is $\Theta(W\sqrt{\frac{p_{rnd}}{n\log n}})$.*
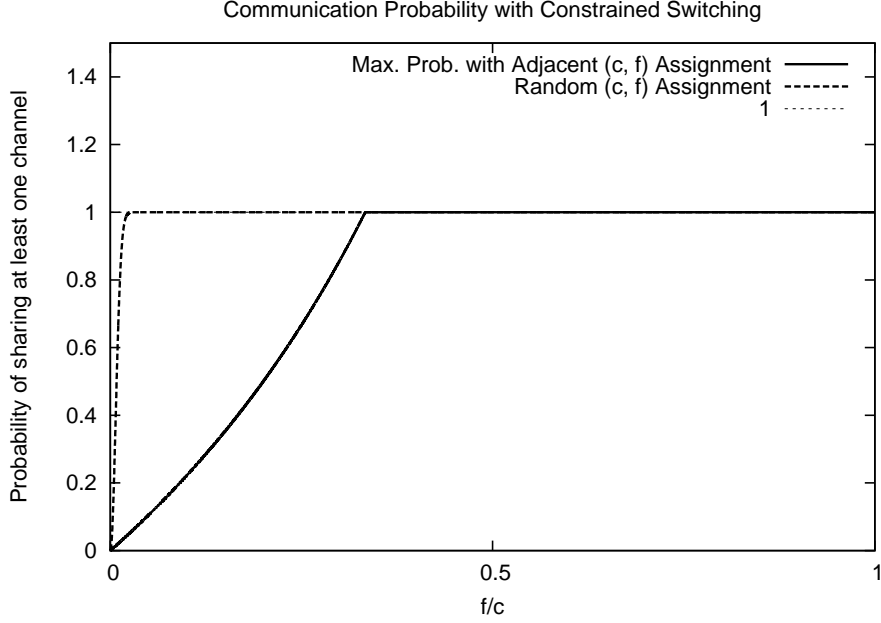
Figure 4.8: Comparison of probability of sharing a channel

4.8 Discussion

We have shown that the capacity for random (c, f) assignment is $\Theta(W\sqrt{\frac{p_{rnd}}{n \log n}})$ in the regime $c = O(\log n)$. It is easy to see that:

$$p_{rnd} = 1 - \left(1 - \frac{f}{c}\right)\left(1 - \frac{f}{c-1}\right)\cdots\left(1 - \frac{f}{c-f+1}\right)$$

note that the product in the R.H.S. above is uniformly 0 whenever $f \geq c - f + 1$, as one of the terms in the product is 0

$$\geq 1 - \left(1 - \frac{f}{c}\right)^f$$

$$\geq 1 - e^{-\frac{f^2}{c}}$$

$$(4.33)$$

Therefore $f = \Omega(\sqrt{c}) \implies p_{rnd} = \Omega(1)$. To illustrate, setting $f = \sqrt{c}$ yields $p_{rnd} \geq 1 - \frac{1}{e} > \frac{1}{2}$. In light of (4.33), our result implies that $f = \Omega(\sqrt{c})$ suffices for achieving capacity of the same order as the unconstrained switching case [65, 66].

We also described a simpler construction that achieves per-flow throughput $\Omega(W\sqrt{\frac{f}{cn \log n}})$. For $f = \sqrt{c}$, using this simpler construction would yield a capacity degradation by a factor of the order of $c^{\frac{1}{4}}$ compared to the unconstrained switching case.

Fig. 4.8 is a numerical plot (obtained by setting c to 10^4, and varying f from 2 to c) depicting how the probability p_{rnd} compares with the probability $p_{adj}^{max} = \min\{\frac{2f-1}{c-f+1}, 1\}$.

Recall that p_{rnd} is the probability that two nodes share at least one channel in random (c, f) assignment, and p_{adj}^{max} is the upper bound on the probability that two nodes share at least one channel in adjacent (c, f) assignment (Chapter 3). It must be remarked that though both models allow nodes to switch between a subset of f channels, the additional degrees of freedom obtained via the random assignment model lead to a much quicker convergence of p_{rnd} toward 1.

It is to be noted that the optimal construction is substantially more complex than the simpler construction and requires that all routes be constructed in lock-step. Thus the two constructions represent an interesting trade-off in capacity versus scheduling/routing complexity.

Moreover, the optimal construction provides many useful insights into the implications of heterogeneous interfaces for routing in a realistic scale network. Note that the need for a synchronized route construction procedure arose from a strong coupling between choices of channels/relays at each hop, over and above what one would find in a network with homogeneous interfaces.

Let us re-examine the implications of heterogeneous interfaces that are subject to switching constraints: if we have to choose a route for a flow, then the first hop transmission must necessarily be scheduled on one of the f channels that the source can switch on (since the source will be sending it); the first relay node must also be one that has at least one channel in common with the source node (so that it can receive the transmission); moreover if channel x is chosen, then the relay node must be capable of switching on channel x. Similarly, the choice of channel at each subsequent hop is limited to the channel-subset of the hop-sender, and the choice of next relay is limited to nodes that can switch on such a channel. Thus the choice of relay at hop i determines the channel choices and consequently relay choices available for hop $i+1$. This leads to a coupling across hops of the same route. Moreover, this also leads to a strong coupling across routes. It is due to these concerns that the capacity achieving construction has a synchronized route selection procedure. We present a simple example to illustrate this issue:

Consider nodes A, B, C, D, X, Y, each of which is equipped with a single interface. Consider two flows $A \rightarrow B$ and $C \rightarrow D$. A, B and C, D are not neighbors, but the nodes X, Y are neighbors of all nodes A, B, C, D, and can thus act as relays for the flows. The channel-
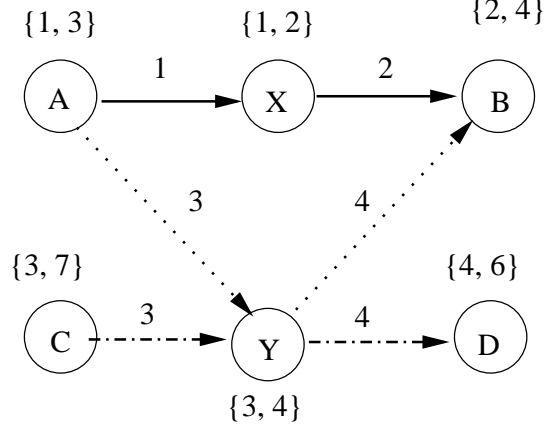
Figure 4.9: Example illustrating coupling between routes

sets of the nodes are as shown in Fig. 4.9. The first flow can use the route $A\xrightarrow{1}X\xrightarrow{2}B$ or $A\xrightarrow{3}Y\xrightarrow{4}B$. The second flow has only one choice $C\xrightarrow{3}Y\xrightarrow{4}D$. Suppose we perform route-selection for the two flows sequentially in the order $A \to B, C \to D$. If the first flow chooses its route without consideration of the second flow and its constraints, it may end up choosing $A\xrightarrow{3}Y\xrightarrow{4}B$. Since the second flow must necessarily choose $C\xrightarrow{3}Y\xrightarrow{4}D$, this will lead to a bottleneck. The optimal choice is for the first flow to use route $A\xrightarrow{1}X\xrightarrow{2}B$ and for the second flow to use $C\xrightarrow{3}Y\xrightarrow{4}D$. If all interfaces could switch on all channels, this problem would not have arisen, as regardless of which route the first flow chose, the second flow could always choose the node-disjoint route, and use different channels on that route. Thus, interfaces with constrained switching ability require more sophisticated routing algorithms to reduce the chances of severe bottleneck formation due to a sub-optimal routing choice.

Chapter 5

Scheduling in Multi-Channel Wireless Networks

In this chapter, we examine scheduling issues in multi-channel wireless networks, where channels may have heterogeneous rate characteristics. We also briefly discuss the scheduling implications of interface heterogeneity. Appropriate scheduling policies are of utmost importance in achieving good throughput characteristics in a multi-hop wireless network. The seminal work of Tassiulas and Ephremides yielded a *throughput-optimal* scheduler, which is capable of scheduling all "feasible" traffic flows while maintaining stability of queues [110]. However, such an optimal scheduler is difficult to implement in practice. Consequently, various imperfect scheduling strategies, which trade-off throughput for simplicity, have been proposed ([75, 119, 120, 103] amongst others).

When multiple orthogonal channels are available in a wireless network, it is possible to get substantial performance improvement (compared to the use of just one of these channels) by harnessing the spectral resource to the maximum extent possible. However, this also gives rise to non-trivial channel coordination issues. The situation is exacerbated by variability in the achievable data-rates across different channels on a link. Such variability may arise due to various reasons, such as the use of different modulations, different propagation characteristics, or time-varying channel conditions. In this chapter, our focus is on heterogeneity in channel rates which is time-invariant.

Computing an optimal schedule, even in a single-channel network, is usually intractable both due to need for global information, and computational complexity. However, imperfect schedulers requiring limited *local* information can typically be designed, which provide acceptable worst-case (and typically much better average case) performance degradation compared to the optimal. In a multi-channel network, the local information exchange required by even an imperfect scheduler can be quite prohibitive, as information may be needed on a per-channel basis. For instance, Lin and Rasool [74] have described a scheduling

algorithm for multi-channel multi-radio wireless networks that requires information about *per-channel* queues at all interfering links. This provides a strong motivation for the study of scheduling algorithms that can operate with limited information, while still providing acceptable worst-case performance guarantees.

In this chapter, we examine the scheduling implications of multiple channels, and heterogeneity in channel-rates. We begin by briefly discussing related work in Section 5.1. We introduce the model, definitions and notation in Section 5.2. Scheduling issues that arise in multi-channel wireless networks are discussed in Section 5.3. Section 5.4 presents a brief summary of our results. We present a result on the cardinality of the set of links scheduled by any maximal scheduler in Section 5.5. In Section 5.6, we derive a lower bound on performance of a *greedy maximal* scheduler, which improves upon existing bounds for this scheduler. In Section 5.7, we describe a scheduler that operates with limited information, and prove a lower bound on its performance. In Section 5.8, we briefly discuss the issue of scheduling with heterogeneous radios, and in Section 5.9 we identify interesting directions for future work.

5.1   Related Work

The issue of throughput-optimal scheduling was considered in the seminal work of Tasiulas and Ephremides [110], in which they described the *Dynamic Backpressure Scheduler*, which is throughput-optimal. The impact of imperfect scheduling on the convergence of joint rate-control and scheduling was examined in [75].

A maximal scheduler combined with local threshold based participation rule has been proposed in [121]. The efficiency ratio of the greedy maximal scheduler has been studied in [25, 49, 50, 48], amongst others. It was shown in [25] that for a class of graphs, with conflicts amongst adjacent links, greedy maximal matching yields an efficiency-ratio of 1. These topologies are those which satisfy a certain property termed the *local pooling condition*. In [49], this was generalized to σ-local pooling ($\sigma \leq 1$), and it was shown that the greedy maximal matching algorithm achieves an efficiency-ratio of σ in all topologies where the local pooling factor is σ. This result was further generalized to general interference models in [50].

A queue-loading algorithm to be used with a maximal scheduler in a multi-channel multi-

radio networks has been described in [74]. Cross-layer resource allocation in multi-channel wireless networks has been considered in [81].

5.2 Preliminaries

We consider a multi-hop wireless network. For simplicity, we largely limit our discussion to nodes equipped with a single radio-interface capable of tuning to any one available channel at any given time. All interfaces in the network have identical operational capabilities, and may switch between the available channels if desired, i.e., there are no switching constraints. Many of the presented results can also be used to obtain results for the case when each node is equipped with multiple interfaces; we briefly discuss this issue.

The wireless network is viewed as a directed graph, with each directed link in the graph representing an available communication link. We model interference using a *conflict* relation between links. Two links are said to conflict with each other if it is only feasible to schedule one of the links on a certain channel at any given time. The conflict relation is assumed to be symmetric. The conflict-based interference model provides a tractable approximation of reality – while it does not capture the wireless channel precisely, it is more amenable to analysis. Such conflict-based interference models have been used frequently in the past work (e.g., [121, 74]).

Time is assumed to be slotted, with the slot duration being 1 unit time (i.e., we use slot duration as the time unit). In each time slot, the scheduler used in the network determines which links should transmit in that time slots, as well as the channel to be used for each such transmission.

We now introduce some notation and terminology.

The network is viewed as a collection of directed links, where each link is a pair of nodes that are capable of direct communication with non-zero rate.

- $\mathcal{L}$ denote the set of directed links in the network.

- $\mathcal{C}$ is the set of all available orthogonal channels. Thus, $|C|$ is the number of available channels.

- We say that a scheduler schedules link-channel pair (l, c) if it schedules link l for transmission on channel c.

- r_l^c denotes the rate achievable on link l by operating link l on channel c, provided that no conflicting link is also scheduled on channel c. For simplicity, we assume that $r_l^c > 0$ for all $l \in \mathcal{L}$ and $c \in \mathcal{C}$ [1]. The rates r_l^c do not vary with time. We also define the following terms: $r_{max} = \max\limits_{l \in \mathcal{L}, c \in \mathcal{C}} r_l^c$, and $r_{min} = \min\limits_{l \in \mathcal{L}, c \in \mathcal{C}} r_l^c$. When two conflicting links are scheduled simultaneously on the same channel, both achieve rate 0.

- β_s denotes the *self-skew-ratio*, defined as the minimum ratio between rates supportable over *different* channels on a *single* link. Therefore, for any two channels c and d, and any link l, we have $\frac{r_l^d}{r_l^c} \geq \beta_s$. Note that $0 < \beta_s \leq 1$.

- β_c denotes the *cross-skew-ratio*, defined as the minimum ratio between rates supportable over the *same* channel on *different* links. Therefore, for any channel c, and any two links l and l': $\frac{r_{l'}^c}{r_l^c} \geq \beta_c$. Note that $0 < \beta_c \leq 1$.

  Let $r_l = \max\limits_{c \in \mathcal{C}} r_l^c$. Let $\sigma_s = \min\limits_{l \in \mathcal{L}} \dfrac{\sum\limits_{c \in \mathcal{C}} r_l^c}{r_l}$. Note that $\sigma_s \geq 1 + \beta_s(\sigma_s - 1)$. Moreover, typically σ_s will be much larger than this worst-case bound. σ_s is largest when $\beta_s = 1$, in which case $\sigma_s = |\mathcal{C}|$.

- $b(l)$ and $e(l)$, respectively, denotes the nodes at the two endpoints of a link. In particular, link l is directed from node $b(l)$ to node $e(l)$.

- $\mathcal{E}(b(l))$ and $\mathcal{E}(e(l))$ denote the set of links incident on nodes $b(l)$ and $e(l)$, respectively. Thus, the links in $\mathcal{E}(b(l))$ and $\mathcal{E}(e(l))$ share an endpoint with link l. Since we focus on single-interface nodes, this implies that if link l is scheduled in a certain time slot, no other link in $\mathcal{E}(b(l))$ or $\mathcal{E}(e(l))$ can be scheduled at the same time. We refer to this as an *interface conflict*. Let $\mathcal{A}(l) = \mathcal{E}(b(l)) \cup \mathcal{E}(e(l))$. Note that $l \in \mathcal{A}(l)$. Links in $\mathcal{A}(l)$ are said to be *adjacent* to link l. Links that have an interface conflict with link l are those that belong to $\mathcal{E}(b(l)) \cup \mathcal{E}(e(l)) \setminus \{l\}$. Let $A_{max} = \max\limits_{l} |\mathcal{A}(l)|$.

- $\mathbf{I}(l)$ denotes the set of links that conflict with link l when scheduled on the same channel. $\mathbf{I}(l)$ may include links that also have an interface-conflict with link l. By convention, l is considered included in $\mathbf{I}(l)$. The subset of $\mathbf{I}(l)$ comprising interfering

---

[1] Though we assume that $r_l^c > 0$ for all l, c, the results can be generalized very easily to handle the case where $r_l^c = 0$ for some link-channel pairs

links that are not adjacent to l is denoted by $\mathbf{I}'(l)$, i.e., $\mathbf{I}'(l) = \mathbf{I}(l) \setminus \mathcal{A}(l)$. Let $I_{max} = \max_l |\mathbf{I}'(l)|$.

- K_l denotes the maximum number of non-adjacent links in $\mathbf{I}'(l)$ that can be scheduled on a given channel simultaneously if l is not scheduled on that channel. $K_l(|\mathcal{C}|)$ denotes the maximum number of non-adjacent links in $\mathbf{I}'(l)$ that can be scheduled simultaneously on any of the $|\mathcal{C}|$ channels (without conflicts) if l is not scheduled for transmission. Note that here we exclude links that have an interface conflict with l.

- K is the largest value of K_l over all links l, i.e., $K = \max_l K_l$. $K_{|\mathcal{C}|}$ is the largest value of $K_l(|\mathcal{C}|)$ over all links l, i.e., $K_{|\mathcal{C}|} = \max_l K_l(|\mathcal{C}|)$. Let $I_{max} = \max_l |\mathbf{I}'(l)|$. It is not hard to see that for *single-interface* nodes:

$$K \leq K_{|\mathcal{C}|} \leq \min\{K|\mathcal{C}|, I_{max}\} \tag{5.1}$$

We remark that the term K as used by us is similar, but not exactly the same as the term K used in [74]. In [74], K denotes the largest number of links that may be scheduled simultaneously if some link l is not scheduled, including links adjacent to l. We exclude the adjacent links in our definition of K. Throughout this text, we will refer to the quantity defined in [74] as κ instead of K.

- Let γ_l be 0 if there are no other links adjacent to l at either endpoint of l, 1 if there are other adjacent links at only one endpoint, and 2 if there are other adjacent links at both endpoints.

- γ is the largest value of γ_l over all links l, i.e., $\gamma = \max_l \gamma_l$.

- *Load vector*: We consider single-hop traffic, i.e., any traffic that originates at a node is destined for a next-hop node, and is transmitted over the link between the two nodes. Under this assumption, all the traffic that must traverse a given link can be treated as a single flow.

  The traffic arrival process for link l is denoted by $\{\lambda(t)\}$. The arrivals in each slot t are assumed i.i.d. with average λ_l. The average load on the network is denoted by *load* vector $\overrightarrow{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_{|\mathcal{L}|}]$, where λ_l denotes the arrival rate for the flow on link l. λ_l may possibly be 0 for some links l.

94

- *Queues*: The packets generated by each flow are first added to a queue maintained at the source node (depending on the algorithm, there could be a single queue for each link, or a queue for each (link, channel) pair).

- *Stability*: The system of queues in the network is said to be stable if, for all queues Q in the network, the following is true:

$$\limsup_{t \to \infty} \frac{1}{t} \sum_{\tau=1}^{t} E[q(\tau)] < \infty \tag{5.2}$$

where $q(\tau)$ denotes the backlog in queue Q at time τ

- *Feasible load vector*: In each time slot, the scheduler used in the network determines which links should transmit and on which channel (recall that each link is a directed link, with a transmitter and a receiver). In different time slots, the scheduler may schedule a different set of links for transmission. A load vector is said to be *feasible*, if there exists a scheduler that can schedule transmissions to achieve stability (as defined above), when using that load vector.

- *Link rate vector*: Depending on the schedule chosen in a given slot by the scheduler, each link l will have a certain transmission rate. For instance, using our notation above, if link l is scheduled to transmit on channel c, it will have rate r_l^c (we assume that, if the scheduler schedules link l on channel c, it does not schedule another conflicting link on that channel). Thus, the *schedule* chosen for a time-slot yields a *link rate vector* for that time slot. Note that *link rate vector* specifies rate of transmission used on each link in a certain time slot. On the other hand, *load vector* specifies the rate at which traffic is generated for each link.

- *Feasible rate region*: The set of all feasible load vectors constitutes the feasible rate-region of the network, and is denoted by Λ. A *throughput-optimal* scheduler is one that is capable of maintaining stable queues for any load vector $\vec{\lambda} \in \Lambda$.

- *Throughput-optimal scheduler:* From the work of [110], it is known that a scheduler that maintains a queue for each link l, and then chooses the schedule given by $\operatorname{argmax}_{\vec{r}} \sum_l q_l r_l$, is throughput-optimal for scenarios with single-hop traffic (q_l is the backlog in link l's queue, and the maximum is taken over all possible link rate vectors

$\overrightarrow{r}$). Note that q_l is a function of time, and queue-backlogs at the start of a time slot are used above for computing the schedule (or link-rate vector) for that slot.

- *Imperfect scheduler*: It is usually difficult to determine the throughput-optimal link-rate allocations, since the problem is typically computationally intractable. Hence, there has been significant recent interest in *imperfect* scheduling policies that can be implemented efficiently. In [75], cross-layer rate-control was studied for an imperfect scheduler that chooses (in each time slot) link-rate vector $\overrightarrow{s}$ such that $\sum_l q_l s_l \geq \delta \operatorname{argmax}_{\overrightarrow{r}} \sum q_l r_l$, for some constant δ ($0 < \delta \leq 1$).

  It was shown [75] that any scheduler with this property can stabilize any load-vector $\overrightarrow{\lambda} \in \delta\Lambda$ – note that if a rate vector $\overrightarrow{\lambda}$ is in Λ, then the rate vector $\delta\overrightarrow{\lambda}$ is in $\delta\Lambda$. $\delta\Lambda$ is also referred to as the *δ-reduced rate-region*. If a scheduler can stabilize all $\overrightarrow{\lambda} \in \delta\Lambda$, its *efficiency-ratio* is said to be δ.

- *Maximal scheduler*: Under our assumed interference model, a schedule is said to be maximal if (a) no two links in the schedule conflict with each other, and (b) it is not possible to add any link to the schedule without creating a conflict (either conflict due to interference, or an interface-conflict).

We utilize the following stability criterion (from [85]) based on Lyapunov drift:

Let $\overrightarrow{U}^{(a)}(t) = (U_i^{(a)}(t))$ be the backlog matrix, where $U_i^{(a)}(t)$ is the backlog in queue i for commodity a. Let $L(\overrightarrow{U})$ be a non-negative function of $\overrightarrow{U}$.

Lemma 42. *(Lyapunov Stability) [85] If the Lyapunov function of unfinished work $L(\overrightarrow{U})$ satisfies:*

$$E[L(\overrightarrow{U}(t+1)) - L(\overrightarrow{U}(t))|\overrightarrow{U}(t)] \leq B - \epsilon \sum_{i,a} \theta_i^{(a)} U_i^{(a)}(t)$$

for some positive constants $B, \theta_i^{(a)}$, then:

$$\limsup_{M\to\infty} \sum_{i,a} \theta_i^{(a)} \left\{ \frac{1}{M} \sum_{k=0}^{M-1} E[U_i^{(a)}(kT)] \right\} \leq B \tag{5.3}$$

Furthermore, if there is a nonzero probability that the system will eventually empty, then a steady state distribution for unfinished work exists, with bounded average occupancies $\overline{U}_i^a$

satisfying

$$\sum_{i,a} \theta_i^{(a)} \overline{U}_i^a \leq B \tag{5.4}$$

We remark that, though the definition of stability used in [85] is different from the definition we use (our assumed definition conforms to *Strong Stability* [37]), the proof of Lemma 42 in [85] establishes stability in the sense of the alternative definition by establishing the condition (5.3), which is equivalent to Strong Stability. Therefore, Lemma 42 can be used for the purpose of our results.

5.3   Scheduling in Multi-channel Wireless Networks

As was discussed previously, throughput-optimal scheduling is often an intractable problem even in a single-channel network. However, imperfect schedulers that achieve a fraction of the stability-region can potentially be implemented in a reasonably efficient manner. Of particular interest is the class of imperfect schedulers know as *maximal schedulers*, which we defined in Section 5.2. The performance of maximal schedulers under various assumptions has been studied in much recent work, e.g., [120, 103], with the focus largely on single-channel wireless networks. The issue of designing a distributed scheduler that approximates a maximal scheduler has been addressed in [51], etc.

When there are multiple channels, but each node has one or few interfaces, an additional degree of complexity is added, in terms of channel selection. In particular, when the link-channel rates r_l^c can be different for different links l, and channels c, the scheduling complexity is exacerbated by the fact that it is not enough to assign different channels to interfering links; for good performance, the channels must be assigned taking achievable rates into account, i.e., individual channel identities are important.

Scheduling in multi-channel multi-radio networks has been examined in [74]. In [74], it was argued that if a simple maximal scheduler is used in such a network, there could possibly be an arbitrary degradation in efficiency-ratio (assuming arbitrary variability in rates) compared to the efficiency-ratio of a maximal scheduler with identical channels. A queue-loading algorithm was been proposed, in conjunction with which, a maximal scheduler can stabilize any vector in $\left(\frac{1}{\kappa+2}\right)\Lambda$, for arbitrary β_c and β_s values. This rule requires knowledge of of the length of queues at all interfering links.
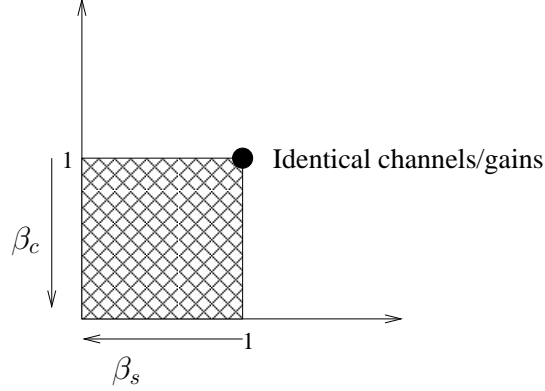
Figure 5.1: 2-D visualization of channel heterogeneity

Variability in channel gains over different links is very much a characteristic of real-world wireless networks, and must indeed be handled by protocols and algorithms. However, if the solutions require extensive information-exchange, the resultant performance improvement may be offset by the increased overhead. In light of this, it is crucial to consider various points of trade-off between information and performance. In this context, the quantities β_s, β_c and σ_s defined in Section 5.2 prove to be useful. The quantities β_s and β_c can be viewed as two orthogonal axes for worst-case channel heterogeneity (Fig. 5.1). The quantity σ_s provides an aggregate (and thus averaged-out) view of heterogeneity along the β_s axis. $\beta_s = 1$ corresponds to a scenario where all channels have identical characteristics, such as bandwidth, modulation/transmission-rate, noise-levels, etc., and the link-gain is a function solely of the separation between sender and receiver. $\beta_c = 1$ corresponds to a scenario where all links have the same sender-receiver separation, and the same conditions/characteristics for any given channel, but the channels may have different characteristics, e.g., an 802.11b channel with a maximum supported data-rate of 11 Mbps, and an 802.11a channel with a maximum supported data-rate of 54 Mbps.

In this chapter, we show that in a single-interface network, a simple maximal scheduler augmented with local traffic-distribution and threshold rules achieves an efficiency-ratio at least $\left(\frac{\sigma_s}{K_{|\mathcal{C}|} + \max\{1,\gamma\}|\mathcal{C}|} \right)$. The noteworthy features of this result are:

1. This scheduler does not require information about queues at interfering links.

2. The performance degradation (compared to the scheduler of [74]) when rates are variable, i.e., $\beta_s, \beta_c \neq 1$, is not arbitrary, and is at worst $\frac{\sigma_s}{|\mathcal{C}|} \geq \frac{1 + \beta_s(|\mathcal{C}| - 1)}{|\mathcal{C}|} \geq \frac{1}{|\mathcal{C}|}$. Thus,
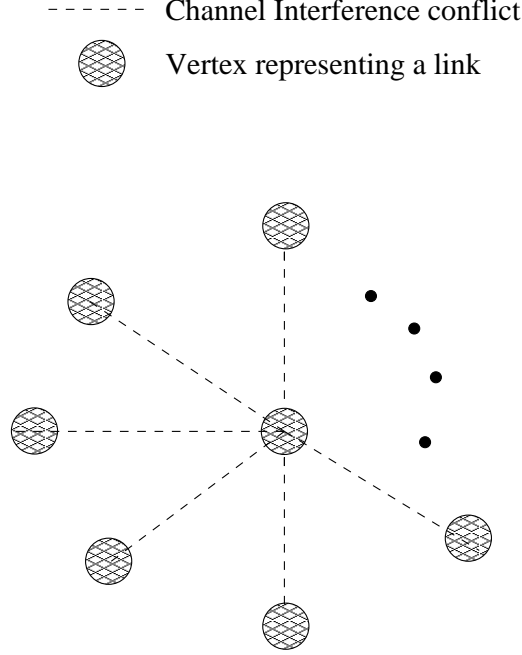
Figure 5.2: Example of improved bound on efficiency ratio: link-interference topology is a star with a center link and x radial links

even with a purely local information based queue-loading rule, it is possible to avoid arbitrary performance degradation even in the worst case. Typically, the performance would be much better.

3. In many network scenarios, the provable lower bound of $\left(\frac{\sigma_s}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|} \right)$ may actually be better than $\frac{1}{\kappa + 2}$. This is particularly likely to happen in networks with single-interface nodes, e.g., suppose we have three channels a, b, c with $r_l^a = 1, r_l^b = 1, r_l^c = 0.5$ for all links l. Then, in the network in Fig. 5.2 (where the link-interference graph is a star with x radial vertices, and there are no interface-conflicts), $K_{|\mathcal{C}|} = x, \gamma = 0, \sigma_s = 2.5$, and we obtain a bound of $\frac{1}{0.4x + 1.2}$, whereas the proved lower bound of the scheduler of [74] is $\frac{1}{x+2}$.

The multi-channel scheduling problem is further complicated if the rates r_l^c are time-varying, i.e., $r_l^c = r_l^c(t)$. However, handling such time-varying rates is beyond the scope of the results in this chapter, and we address only the case where rates do not exhibit time-variation.

5.4 Summary of Results

For multi-channel wireless networks with single-interface nodes, we present lower bounds on the efficiency-ratio of a class of maximal schedulers (including both centralized and distributed schedulers), which indicate that the worst-case efficiency-ratio can be higher when there are multiple channels (as compared to the single-channel case). More specifically, we show that:

- The number of links scheduled by any maximal scheduler are within at least a δ fraction of the maximum number of links activated by any feasible schedule, where:

$$\delta = \max\left\{\frac{|\mathcal{C}|}{K_{|\mathcal{C}|} + \max\{1,\gamma\}|\mathcal{C}|}, \frac{1}{\max\{1,K+\gamma\}}\right\}$$

- A centralized greedy maximal (CGM) scheduler achieves an efficiency-ratio at least $\max\{\frac{\sigma_s}{K_{|\mathcal{C}|}+\max\{1,\gamma\}|\mathcal{C}|}, \frac{1}{\max\{1,K+\gamma\}}\}$ This constitutes an improvement over the lower bound for the CGM scheduler proved in [74]. Since $K_{|\mathcal{C}|} \leq \min\{K|\mathcal{C}|, I_{max}\} \leq \kappa|\mathcal{C}|$, this new bound on efficiency-ratio can often be substantially tighter.

- We show that any maximal scheduler, in conjunction with a simple local queue-loading rule, and a threshold-based link-participation rule, achieves an efficiency-ratio of at least $\left(\frac{\sigma_s}{K_{|\mathcal{C}|}+\max\{1,\gamma\}|\mathcal{C}|}\right)$. This scheduler is of significant interest as it does not require information about queues at all interfering links.

Note that the text below makes the natural assumption that two links that conflict with each other (due to interference or interface-conflict) are **not** scheduled in the same timeslot by any scheduler discussed in the rest of this chapter.

5.5 Maximal Schedulers

We begin by proving a result about the cardinality of the set of links scheduled by any maximal scheduler.

Theorem 10. *Let $\mathcal{S}_{opt}$ denote the set of links scheduled by a scheduler that seeks to maximize the* number *of links scheduled for transmission, and let $\mathcal{S}_{max}$ denote the set of links*

activated by any *maximal scheduler. Then the following is true:*

$$|\mathcal{S}_{max}| \geq \max\{\frac{|\mathcal{C}|}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}, \frac{1}{\max\{1, K + \gamma\}}\}|\mathcal{S}_{opt}| \tag{5.5}$$

Proof. Denote by $c^m(l')$ the channel on which a link l' is scheduled in $\mathcal{S}_{max}$.

Consider $l \in \mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}$. Since l was not scheduled by the maximal scheduler, this implies that at least one of the following events must be true:

1. Condition 1: $\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}} \cap \mathcal{A}(l) \neq \phi$.

2. Condition 2: For each channel $c \in \mathcal{C}$, there exists some link $l'_c \in \mathcal{S}_{max} \cap \mathbf{I}'(l)$, such that $c^m(l'_c) = c$.

Now, define sets A_{if} and A_{in} as follows:

$\mathcal{A}_{if} = \{l : l \in \mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}$ and Condition 1 holds$\}$

$\mathcal{A}_{in} = (\mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}) \setminus \mathcal{A}_{if}$

Thus $\mathcal{A}_{if}$ comprises the set of links in $\mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}$ that have an interface conflict with some link in the maximal-schedule, while $\mathcal{A}_{in}$ comprises the set of links in $\mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}$ that are blocked in the maximal-schedule purely by channel-interference conflicts.

For each $l \in \mathcal{A}_{in}$, let $\mathcal{Y}_l = \mathcal{S}_{max} \cap \mathbf{I}'(l)$. Taking note of Condition 2, each link $l \in \mathcal{A}_{in}$ must be blocked on each channel $c \in \mathcal{C}$ by at least one link in $\mathcal{Y}_l$. Any link $l' \in \mathcal{S}_{max}$ can occur in the $\mathcal{Y}_l$ of at most $K_{|\mathcal{C}|}$ non-adjacent links $l \in \mathcal{S}_{opt}$.

Therefore, it follows that:

$$|\mathcal{C}||\mathcal{A}_{in}| \leq K_{|\mathcal{C}|} |\mathcal{S}_{max}| \tag{5.6}$$

Any interface-conflicts experienced by links in $\mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}$ must necessarily be caused by links in $\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}}$. Since a link can only block up to γ links through interface-conflicts, we obtain that:

$$|\mathcal{A}_{if}| \leq \gamma |\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}}| \tag{5.7}$$

Thus we obtain the following:

$$
\begin{aligned}
\frac{|\mathcal{S}_{opt}|}{|\mathcal{S}_{max}|} &= \frac{|\mathcal{S}_{max} \cap \mathcal{S}_{opt}| + |\mathcal{S}_{opt} \cap \overline{\mathcal{S}_{max}}|}{|\mathcal{S}_{max}|} = \frac{|\mathcal{S}_{max} \cap \mathcal{S}_{opt}| + |\mathcal{A}_{if}| + |\mathcal{A}_{in}|}{|\mathcal{S}_{max}|} \\
&\leq \frac{|\mathcal{S}_{max} \cap \mathcal{S}_{opt}| + \gamma|\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}}| + \frac{K_{|\mathcal{C}|}}{|\mathcal{C}|}|\mathcal{S}_{max}|}{|\mathcal{S}_{max}|} \quad \text{from (5.7) \& (5.6)} \\
&= \frac{|\mathcal{S}_{max} \cap \mathcal{S}_{opt}| + |\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}}| + (\gamma-1)|\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}}| + \frac{K_{|\mathcal{C}|}}{c}|\mathcal{S}_{max}|}{|\mathcal{S}_{max}|} \\
&= \frac{|\mathcal{S}_{max}| + (\gamma-1)|\mathcal{S}_{max} \cap \overline{\mathcal{S}_{opt}}| + \frac{K_{|\mathcal{C}|}}{|\mathcal{C}|}|\mathcal{S}_{max}|}{|\mathcal{S}_{max}|} \\
&\leq \frac{|\mathcal{S}_{max}| + \max\{0, \gamma-1\}|\mathcal{S}_{max}| + \frac{K_{|\mathcal{C}|}}{|\mathcal{C}|}|\mathcal{S}_{max}|}{|\mathcal{S}_{max}|} \\
&= 1 + \max\{0, \gamma-1\} + \frac{K_{|\mathcal{C}|}}{|\mathcal{C}|} \\
&= \max\{1, \gamma\} + \frac{K_{|\mathcal{C}|}}{|\mathcal{C}|}
\end{aligned}
\tag{5.8}
$$

Furthermore, consider any link l in $\mathcal{S}_{max}$. Either l is scheduled even in $\mathcal{S}_{opt}$, or if l is not scheduled in $\mathcal{S}_{opt}$, at most K links in $\mathbf{I}'(l)$, and γ links in $\mathcal{A}(l) \setminus \{l\}$ could have been scheduled in $\mathcal{S}_{opt}$. Thus:

$$
\frac{|\mathcal{S}_{opt}|}{|\mathcal{S}_{max}|} \leq \max\{1, K + \gamma\}
\tag{5.9}
$$

Combining (5.8) and (5.9), we obtain that:

$$
|\mathcal{S}_{max}| \geq \max\left\{ \frac{|\mathcal{C}|}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}, \frac{1}{\max\{1, K + \gamma\}} \right\} |\mathcal{S}_{opt}|
\tag{5.10}
$$

$\square$

5.6 Centralized Greedy Maximal Scheduler

A centralized greedy maximal (CGM) scheduler operates in the manner described below.

In each timeslot:

1. Calculate link weights $w_l^c = q_l r_l^c$ for all links l and channels c.

2. Sort the link-channel pairs (l, c) in non-increasing order of w_l^c.

3. Add the first link-channel pair in the sorted list (i.e., the one with highest weight) to

the schedule for the timeslot, and remove from the list all link-channel pairs that are no longer feasible (due to either interface or interference conflicts).

4. Repeat step 3 until the list is exhausted (i.e., no more links can be added to the schedule).

In [74], it was shown that this centralized greedy maximal (CGM) scheduler can achieve an approximation-ratio at least $\left(\frac{1}{\kappa+2}\right)$ in a multi-channel multi-radio network, where κ is the maximum number of links conflicting with a link l that may possibly be scheduled concurrently when l is not scheduled. This bound holds for arbitrary values of β_s and β_c, and variable number of interfaces per node.

However, this bound can be quite loose in multi-channel wireless networks where each device has one or few interfaces.

In this section, we prove an improved bound on the efficiency-ratio achievable with the CGM scheduler for *single-interface* nodes. We also briefly discuss how it can be used to obtain a bound for multi-interface nodes.

Theorem 11. *Let $\mathcal{S}_{opt}$ denote the set of links activated by an optimal scheduler that chooses a set of link-channel pairs (l, c) for transmission such that $\sum w_l^c$ is maximized. Let $c^*(l)$ denote the channel assigned to link $l \in \mathcal{S}_{opt}$ by this optimal scheduler.*

Let $\mathcal{S}_g$ denote the set of links activated by the centralized greedy maximal (CGM) scheduler, and let $c^g(l)$ denote the channel assigned to a link $l \in \mathcal{S}_g$.

Then:

$$\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} \geq \max\left\{\frac{\sigma_s}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}, \frac{1}{\max\{1, K + \gamma\}}\right\} \sum_{l \in \mathcal{S}_{opt}} w_l^{c^*(l)} \qquad (5.11)$$

Proof. We denote by $c^*(l)$ the channel on which $l \in \mathcal{S}_{opt}$ is activated by the optimal scheduler. $c^g(l)$ is the channel on which $l \in \mathcal{S}_g$ is activated by the CGM scheduler. If a link l is not in $\mathcal{S}_{opt}$ or $\mathcal{S}_g$, then, as a matter of notational convention, it can be said that $c^*(l) = \perp$ or $c^g(l) = \perp$ respectively, where $\perp$ denoted "undefined".

Consider $l \in \mathcal{S}_{opt} \cap \overline{\mathcal{S}_g}$. Therefore, l was not scheduled by the CGM scheduler. This implies that during some step k of the execution of the CGM algorithm, l's status changed from schedulable to unschedulable. This could happen for one of two reasons: (1) in step k, some link l' incident on one of l's endpoints was selected by the CGM scheduler,

thereby making l unschedulable due to an interface-conflict (2) in step k, all c channels became infeasible for l to be scheduled, implying that for all $c \in \mathcal{C}$, some link $l' \in \mathbf{I}'(l)$ was scheduled on c by the scheduler by the end of step k.

By the definition of the CGM scheduler, a link l' would be preferentially selected for scheduling over l (while l was still schedulable) only if the resultant weight contribution $w_{l'}^{c^g(l')}$ equals or exceeds the best weight that could be achieved by scheduling l on some still feasible channel. Thus, at least one of the following two conditions must be true:

1. Condition 1: There exists a link $l' \in \mathcal{S}_g \cap \overline{\mathcal{S}_{opt}} \cap \mathcal{A}(l)$ such that $w_{l'}^{c^g(l')} \geq w_l^c$ for at least one channel $c \in \mathcal{C}$.

2. Condition 2: For each channel $c \in \mathcal{C}$, there exists some link $l'_c \in \mathcal{S}_g \cap \mathbf{I}'(l)$ such that $w_{l'_c}^c \geq w_l^c$.

Now, define sets A_{if} and A_{in} as follows:

$\mathcal{A}_{if} = \{l : l \in \mathcal{S}_{opt} \cap \overline{\mathcal{S}_g} \quad \text{and Condition 1 holds}\}$.

$\mathcal{A}_{in} = (\mathcal{S}_{opt} \cap \overline{\mathcal{S}_g}) \setminus \mathcal{A}_{if}$

Let $\mathcal{S}_{b,m} = \{l : l \in \mathcal{S}_g \cap \mathcal{S}_{opt}, w_l^{c^g(l)} \geq w_l^{c^*(l)}\}$

Let $\mathcal{S}_{b,s} = \{l : l \in \mathcal{S}_g \cap \mathcal{S}_{opt}, w_l^{c^g(l)} < w_l^{c^*(l)}\}$

Then $\mathcal{S}_{b,m}$ and $\mathcal{S}_{b,s}$ constitute a partition of $\mathcal{S}_g \cap \mathcal{S}_{opt}$.

Define two subsets of $\mathcal{A}_{if}$ as follows:

$$\mathcal{A}_{if,1} = \{l : l \in \mathcal{A}_{if}, c^*(l) \text{ was not available to } l \text{ when } l\text{'s first interface}$$
$$\text{got used up during CGM scheduling}\}$$

$$\mathcal{A}_{if,2} = \{l : l \in \mathcal{A}_{if}, c^*(l) \text{ was still available to } l \text{ when } l\text{'s first interface}$$
$$\text{got used up during CGM scheduling}\}$$

From the centralized greedy nature of the scheduler, if a link $l' \in \mathbf{I}'(l)$ was scheduled on some $c \in \mathcal{C}$ in $\mathcal{S}_g$ while l was still schedulable on some subset of channels $\mathcal{D} \subseteq \mathcal{C}$, this implies that $w_{l'}^c \geq w_l^d$ for all $d \in \mathcal{D}$.

It is true that at the time when $l \in \mathcal{S}_{b,s}$ was assigned $c^g(l)$, all other $c \in \mathcal{C}$ with $r_l^c > r_l^{c^g(l)}$ were already assigned to some other $l' \in \mathbf{I}'(l)$, with $w_{l'}^{c^g(l')} = w_{l'}^c \geq w_l^c$. Therefore, if $\mathcal{D}_l^g$

is the set of channels on which l was still schedulable when l was chosen for scheduling on $c^g(l)$, then: $\forall\, d \in \mathcal{D}_l^g : r_l^d \leq r_l^{c^g(l)}$, and $|\mathcal{D}_l^g| \leq |\mathcal{C}| - 1$ since $c^*(l) \notin \mathcal{D}_l^g$.

Therefore for each $l \in \mathcal{S}_{b,s}$:

$$\sum_{c \in \mathcal{C} \setminus \mathcal{D}_l^g} \sum_{\substack{l' \in \mathbf{I}'(l) \\ c^g(l')=c}} w_{l'}^{c^g(l')} \geq \sum_{c \in \mathcal{C}} w_l^c - \sum_{d \in \mathcal{D}_l^g} w_l^d \geq \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1) w_l^{c^g(l)} \tag{5.12}$$

Let $\mathcal{B}_1(l) = \{ l' | l' \in (\mathcal{S}_g \cap \mathbf{I}'(l)), c^g(l') \in \mathcal{C} \setminus \mathcal{D}_l^g \}$.

$$\therefore \sum_{l \in \mathcal{S}_{b,s}} \left(\sum_{c \in \mathcal{C} \setminus \mathcal{D}_l^g} \sum_{\substack{l' \in \mathbf{I}'(l) \\ c^g(l')=c}} w_{l'}^{c^g(l')} \right) \geq \sum_{l \in \mathcal{S}_{b,s}} \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1) \sum_{l \in \mathcal{S}_{b,s}} w_l^{c^g(l)}$$

$$\therefore \sum_{l \in \mathcal{S}_{b,s}} \left(\sum_{l' \in \mathcal{B}_1(l)} w_{l'}^{c^g(l')} \right) \geq \sum_{l \in \mathcal{S}_{b,s}} \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1) \sum_{l \in \mathcal{S}_{b,s}} w_l^{c^g(l)} \tag{5.13}$$

We now consider links $l \in \mathcal{A}_{if}$.

Let us denote by $f(l)$ the link l' in $\mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}$ that is the cause of blocking the first interface of link $l \in \mathcal{A}_{if}$, i.e., $f(l)$ is the link that first caused l to experience an interface-conflict.

We first consider links $l \in \mathcal{A}_{if,1}$:

It is true that if $f(l) = l' \in \mathcal{A}(l) \cap (\mathcal{S}_g \cap \overline{\mathcal{S}_{opt}})$ was assigned a channel $c^g(l')$ in $\mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}$ while $l \in \mathcal{A}_{if,1}$ was still schedulable on some subset of channels $\mathcal{D}_l \subseteq \mathcal{C} \setminus \{c^*(l)\}$ then $w_{l'}^{c^g(l')} \geq w_l^d$ for all $d \in \mathcal{D}_l$, and $|\mathcal{D}_l| \leq |\mathcal{C}| - 1$ since $c^*(l) \notin \mathcal{D}_l$ (note that $c^* \notin \mathcal{D}_l$ by the definition of $\mathcal{A}_{if,1}$).

Let $B = \sum_{l \in \mathcal{A}_{if,1}} w_{f(l)}^{c^g(f(l))}$.

Furthermore, at least one link $l' \in \mathbf{I}'(l)$ was scheduled on each $c \in \mathcal{C} \setminus \mathcal{D}_l$, and for each such c, l', it is evident that $w_{l'}^{c^g(l')} = w_{l'}^c \geq w_l^c$ (since channels in $\mathcal{C} \setminus \mathcal{D}_l$ were no longer feasible for l at the time its first interface got used up). This yields:

$$\sum_{c \in \mathcal{C} \setminus \mathcal{D}_l} \sum_{\substack{l' \in \mathbf{I}'(l) \\ c^g(l')=c}} w_{l'}^{c^g(l')} \geq \sum_{c \in \mathcal{C}} w_l^c - \sum_{d \in \mathcal{D}_l} w_l^d \geq \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1) w_{f(l)}^{c^g(f(l))} \tag{5.14}$$

Resultantly:

$$\sum_{l \in \mathcal{A}_{if,1}} \left(\sum_{c \in \mathcal{C} \setminus \mathcal{D}_l} \sum_{\substack{l' \in \mathbf{I}'(l) \\ c^g(l')=c}} w_{l'}^{c^g(l')} \right) \geq \sum_{l \in \mathcal{A}_{if,1}} \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1)B \tag{5.15}$$

Let $\mathcal{B}_2(l) = \{l' | l' \in (\mathcal{S}_g \cap \mathbf{I}'(l)), c^g(l') \in \mathcal{C} \setminus \mathcal{D}_l\}$.

$$\therefore \sum_{l \in \mathcal{A}_{if,1}} \left(\sum_{l' \in \mathcal{B}_2(l)} w_{l'}^{c^g(l')} \right) \geq \sum_{l \in \mathcal{A}_{if,1}} \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1)B \tag{5.16}$$

We next consider links $l \in \mathcal{A}_{if,2}$:

From the definition of $\mathcal{A}_{if,2}$, for each link $l \in \mathcal{A}_{if,2}$, some link $f(l) = l'$ adjacent to l was scheduled in $\mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}$ at a time when l was still schedulable on $c^*(l)$. This implies that $w_{l'}^{c^g(l')} \geq w_l^{c^*(l)}$. Let $E = \sum_{l \in \mathcal{A}_{if,2}} w_{f(l)}^{c^g(f(l))}$ (recall the definition of $f(l)$ for links $l \in \mathcal{A}_{if}$).

Thus we obtain:

$$B + \sum_{l \in \mathcal{A}_{if,2}} w_l^{c^*(l)} \leq B + E \leq \gamma \sum_{l \in \mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}} w_l^{c^g(l)}$$

$$\therefore \sum_{l \in \mathcal{A}_{if,2}} w_l^{c^*(l)} \leq \gamma \sum_{l \in \mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} - B \tag{5.17}$$

We now consider links $l \in \mathcal{A}_{in}$:

From the definition of A_{in}, it follows that for each $c \in \mathcal{C}$, there is at least one $l' \in \mathbf{I}'(l)$ scheduled on c such that $w_{l'}^{c^g(l')} = w_{l'}^c \geq w_l^c$. Given $l \in \mathcal{A}_{in}$, let $\mathcal{B}_3(l) = \mathcal{S}_g \cap \mathbf{I}'(l)$. Then:

$$\sum_{l \in \mathcal{A}_{in}} \left(\sum_{l' \in \mathcal{B}_3(l)} w_{l'}^{c^g(l')} \right) \geq \sum_{l \in \mathcal{A}_{in}} \sum_{c \in \mathcal{C}} w_l^c \tag{5.18}$$

Also note that for any link $l' \in \mathcal{S}_g$, at most $K_{|\mathcal{C}|}$ links in $\mathbf{I}'_{l'}$ can be scheduled in $\mathcal{S}_{opt}$. Thus, any link $l' \in \mathcal{S}_g$ figures in $\mathcal{B}_1(l)$ or $\mathcal{B}_2(l)$ or $\mathcal{B}_3(l)$ of at most $K_{|\mathcal{C}|}$ links $l \in \mathcal{S}_{opt}$.

In light of this observation, the definition of σ_s, and using (5.13), (5.16) and (5.18):

$$\sum_{l \in \mathcal{S}_{b,s}} \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1) \sum_{l \in \mathcal{S}_{b,s}} w_l^{c^g(l)} + \sum_{l \in \mathcal{A}_{if,1}} \sum_{c \in \mathcal{C}} w_l^c - (|\mathcal{C}| - 1)B + \sum_{l \in \mathcal{A}_{in}} \sum_{c \in \mathcal{C}} w_l^c \leq K_{|\mathcal{C}|} \sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} \tag{5.19}$$

106

Rearranging and noting that $\sum_{c \in \mathcal{C}} w_l^c \geq \sigma_s w_l^{c^*(l)}$:

$$\sigma_s \left(\sum_{l \in \mathcal{S}_{b,s}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{if,1}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{in}} w_l^{c^*(l)} \right) \leq K_{|\mathcal{C}|} \sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} + (|\mathcal{C}| - 1) \left(\sum_{l \in \mathcal{S}_{b,s}} w_l^{c^g(l)} + B \right)$$

$$\therefore \sum_{l \in \mathcal{S}_{b,s}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{if,1}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{in}} w_l^{c^*(l)} \leq \frac{K_{|\mathcal{C}|}}{\sigma_s} \sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} + \frac{|\mathcal{C}| - 1}{\sigma_s} \left(\sum_{l \in \mathcal{S}_{b,s}} w_l^{c^g(l)} + B \right)$$
(5.20)

This yields the following:

$$\frac{\sum_{l \in \mathcal{S}_{opt}} w_l^{c^*(l)}}{\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}} = \frac{\sum_{l \in \mathcal{S}_g \cap \mathcal{S}_{opt}} w_l^{c^*(l)} + \sum_{l \in \mathcal{S}_{opt} \cap \overline{\mathcal{S}_g}} w_l^{c^*(l)}}{\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}}$$

$$= \frac{\sum_{l \in \mathcal{S}_{b,m}} w_l^{c^*(l)} + \sum_{l \in \mathcal{S}_{b,s}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{if,1}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{if,2}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{in}} w_l^{c^*(l)}}{\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}}$$

$$= \frac{\sum_{l \in \mathcal{S}_{b,m}} w_l^{c^*(l)} + \left(\sum_{l \in \mathcal{S}_{b,s}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{if,1}} w_l^{c^*(l)} + \sum_{l \in \mathcal{A}_{in}} w_l^{c^*(l)} \right) + \sum_{l \in \mathcal{A}_{if,2}} w_l^{c^*(l)}}{\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}}$$

$$\leq \frac{1}{\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}} \left(\sum_{l \in \mathcal{S}_{b,m}} w_l^{c^g(l)} + \frac{K_{|\mathcal{C}|} \sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} + (|\mathcal{C}| - 1) \left(\sum_{l \in \mathcal{S}_{b,s}} w_l^{c^g(l)} + B \right)}{\sigma_s} + \gamma \sum_{l \in \mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} - B \right)$$

from (5.20), (5.17)

$$\leq \frac{1}{\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}} \left(\sum_{l \in \mathcal{S}_{b,m}} w_l^{c^g(l)} + \frac{\sigma_s (\gamma \sum_{l \in \mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} - B)}{\sigma_s} \right.$$

$$\left. + \frac{K_{|\mathcal{C}|} \sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} + (|\mathcal{C}| - 1) \left(\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} - \sum_{l \in \mathcal{S}_{b,m}} w_l^{c^g(l)} - \sum_{l \in \mathcal{S}_g \cap \overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} + B \right)}{\sigma_s} \right)$$

$$
\leq \frac{1}{\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)}} \left(\frac{|\mathcal{C}|}{\sigma_s}\sum_{l\in\mathcal{S}_{b,m}} w_l^{c^g(l)} + \frac{|\mathcal{C}|\gamma\displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} - (|\mathcal{C}|-1)B}{\sigma_s} \right.
$$

$$
\left. + \frac{K_{|\mathcal{C}|}\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)} + (|\mathcal{C}|-1)\left(\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)} - \displaystyle\sum_{l\in\mathcal{S}_{b,m}} w_l^{c^g(l)} - \displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} + B\right)}{\sigma_s} \right)
$$

noting that $\gamma\displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} - B \geq 0$

$$
\leq \frac{1}{\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)}} \left(\frac{(|\mathcal{C}|-1)\displaystyle\sum_{l\in\mathcal{S}_{b,m}} w_l^{c^g(l)} + \displaystyle\sum_{l\in\mathcal{S}_{b,m}} w_l^{c^g(l)} + K_{|\mathcal{C}|}\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)}}{\sigma_s} + \frac{\gamma\displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)}}{\sigma_s} \right.
$$

$$
\left. + \frac{(|\mathcal{C}|-1)\left(\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)} - \displaystyle\sum_{l\in\mathcal{S}_{b,m}} w_l^{c^g(l)} - \displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} + B + \gamma\displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)} - B\right)}{\sigma_s} \right)
$$

$$
\tag{5.21}
$$

$$
\leq \left(\frac{K_{|\mathcal{C}|}\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)} + (|\mathcal{C}|-1)\left(\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)} + (\gamma-1)\displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)}\right)}{\sigma_s\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)}} \right.
$$

$$
\left. + \frac{\left(\displaystyle\sum_{l\in\mathcal{S}_{b,m}} w_l^{c^g(l)} + \gamma\displaystyle\sum_{l\in\mathcal{S}_g\cap\overline{\mathcal{S}_{opt}}} w_l^{c^g(l)}\right)}{\sigma_s\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)}} \right)
$$

$$
\leq \frac{K_{|\mathcal{C}|} + (|\mathcal{C}|-1)(1+\max\{0,\gamma-1\}) + \max\{1,\gamma\}}{\sigma_s}
$$

$$
= \frac{K_{|\mathcal{C}|} + \max\{1,\gamma\}|\mathcal{C}|}{\sigma_s}
\tag{5.22}
$$

Thus $\displaystyle\sum_{l\in\mathcal{S}_g} w_l^{c^g(l)} \geq \frac{\sigma_s}{K_{|\mathcal{C}|}+\max\{1,\gamma\}|\mathcal{C}|}\displaystyle\sum_{l\in\mathcal{S}_{opt}} w_l^{c^*(l)}$. When $\beta_s = 1$, this reduces to a ratio of $\frac{|\mathcal{C}|}{K_{|\mathcal{C}|}+\max\{1,\gamma\}|\mathcal{C}|}$.

We now prove another bound by showing that:

$$\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)} \geq \frac{1}{\max\{1, K + \gamma\}} \sum_{l \in \mathcal{S}_{opt}} w_l^{c^*(l)} \qquad (5.23)$$

This is obtained via an argument very similar to that used in [74] to prove a bound of $\left(\frac{1}{\kappa + 2}\right)$ for the CGM scheduler, except that we refine the analysis based on a more precise characterization of the interference topology:

Consider any link l in $\mathcal{S}_{opt}$. Either l is scheduled on $c^*(l)$ even in $\mathcal{S}_g$, or if l is not scheduled on $c^*(l)$, then either (1) some link $l' \in \mathbf{I}'(l)$ must be scheduled on $c^*(l)$ in $\mathcal{S}_g$ (i.e., $c^g(l') = c^*(l)$), such that $w_{l'}^{c^g(l')} \geq w_l^{c^*(l)}$, or (2) some link $l' \in \mathcal{A}(l) \setminus \{l\}$ must be scheduled on some channel $c^g(l')$ such that $w_{l'}^{c^g(l')} \geq w_l^{c^*(l)}$. However, any link $l' \in \mathcal{S}_g$ can only have pure interference conflict with at most K links that were scheduled in $\mathcal{S}_{opt}$ on that channel, and interface conflict with at most γ links in $\mathcal{A}(l) \cap \mathcal{S}_{opt}$. Thus:

$$\frac{\displaystyle\sum_{l \in \mathcal{S}_{opt}} w_l^{c^*(l)}}{\displaystyle\sum_{l \in \mathcal{S}_g} w_l^{c^g(l)}} \leq \max\{1, K + \gamma\} \qquad (5.24)$$

Combining (5.21) and (5.24) yields the result. $\qquad \square$

Theorem 11 leads to the following result:

Theorem 12. *The centralized greedy maximal (CGM) scheduler can stabilize the δ-reduced rate-region, where:*

$$\delta = \max \left\{ \frac{\sigma_s}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}, \frac{1}{\max\{1, K + \gamma\}} \right\}$$

Proof. We earlier discussed a result from [75] that any scheduler, which chooses rate-allocation $\overrightarrow{s}$ such that $\sum q_l s_l \geq \delta \operatorname{argmax} \sum q_l r_l$, can stabilize the δ-reduced rate-region. Using Theorem 11 and this result, we obtain the above result. $\qquad \square$

We remark that the above bound is independent of β_c.

5.6.1 Extension to Multiple Interfaces per Node

We now describe how the result can be extended to networks where each node may have more than one interface.

Given the original network *node-graph* $G = (V, E)$, construct the following transformed graph $G' = (V', E')$:

For each node $v \in V$, if v has m_v interfaces, create m_v nodes $v_1, v_2, ...v_{m_v}$ in V'. For each edge $(u, v) \in E$, where u, v have m_u, m_v interfaces respectively, create edges $(u_i, v_j), 1 \leq i \leq m_u, 1 \leq j \leq m_v$, and set $q_{(u_i, v_j)} = q_{(u,v)}$. Set the achievable channel rate appropriately for each edge in E' and each channel. For example, assuming that the channel-rate is solely a function of u, v and c, then: for each channel c, set $r^c_{(u_i, v_j)} = r^c_{(u,v)}$.

The transformed graph G' comprises only single-interface links, and thus Theorem 11 applies to it. Moreover, it is not hard to see that a schedule that maximizes $\sum q_l r_l$ in G' also maximizes $\sum q_l r_l$ in G. Thus, the efficiency-ratio from Theorem 11 for network graph G' yields an efficiency-ratio for the performance of the CGM scheduler in the multi-interface network.

We briefly touch upon how one would expect the ratio to vary as the number of interfaces at each node increases. Note that the efficiency-ratio depends on $\beta_s, |\mathcal{C}|, K_{|\mathcal{C}|}, \gamma$. Of these β_s and $|\mathcal{C}|$ are always the same for both G and G'. γ is also always the same for any G' derived from a given node-graph G, as it depends only on the number of other node-links incident on either endpoint of a node-link in G (which is a property of the node topology, and not the number of interfaces each node has). However, $K_{|\mathcal{C}|}$ might potentially increase in G' as there are many more non-adjacent interfering *links* when each interface is viewed as a distinct node. Thus, for a given number of channels $|\mathcal{C}|$, one would expect the provable efficiency-ratio to initially decrease as we add more interfaces, and then become static.

While this may initially seem counter-intuitive, this is explained by the observation that multiple orthogonal channels yielded a better efficiency-ratio in the single-interface case since there was more spectral resource, but limited hardware (interfaces) to utilize it. Thus, the additional channels could be effectively used to alleviate the impact of sub-optimal scheduling. When the hardware is commensurate with the number of channels, the situation (compared to an optimal scheduler) increasingly starts to resemble a single-channel single-interface network.

5.6.2 The Special Case of $|\mathcal{C}|$ Interfaces per Node

Let us consider the special case where each node in the network has $|\mathcal{C}|$ interfaces, and achievable rate on a link between nodes u, v and all channels $c \in \mathcal{C}$ is solely a function of u, v and c (and not of the interfaces used). In this case, it is possible to obtain a simpler transformation. Given the original network node-graph $G = (V, E)$, construct $|\mathcal{C}|$ copies of this graph, viz., $G_1, G_2, ..., G_{|\mathcal{C}|}$, and view each node in each graph as having a single-interface, and each network as having access to a single channel. Then each network graph G_i can be viewed in isolation, and the throughput obtained in the original graph is the sum of the throughputs in each graph. From Theorem 11, in each graph we can show that the CGM scheduler is within $\left(\frac{1}{\max\{1, K+\gamma\}} \right) = \min\{1, \frac{1}{K+\gamma}\}$ of the optimal. Thus, even in the overall network, the CGM scheduler is within $\min\{1, \frac{1}{K+\gamma}\}$ of the optimal.

5.7 A Rate-Proportional Maximal Multi-Channel (RPMMC) Scheduler

In this section, we describe a scheduler where a link does not require any information about queue-lengths at interfering links.

The set of all links in denoted by $\mathcal{L}$. The arrival process for link l is i.i.d. over all time-slots t, and is denoted by $\{\lambda_l(t)\}$, with $E[\lambda_l(t)] = \lambda_l$. We make no assumption about independence of arrival processes for two links l, k. However, we consider only the class of arrival processes for which $E[\lambda_l(t)\lambda_k(t)]$ is bounded, i.e., $E[\lambda_l(t)\lambda_k(t)] \leq \eta$ for all $l \in \mathcal{L}, k \in \mathcal{L}$, where η is a suitable constant.

Consider the following scheduler:

Rate-Proportional Maximal Multi-Channel (RPMMC) Scheduler

Each link maintains a queue for each channel. The length of the queue for link l and channel c at time t is denoted by $q_l^c(t)$. In time-slot t: only those link-channel pairs with $q_l^c(t) \geq r_l^c$ participate, and the scheduler computes a maximal schedule from amongst the participating links. The new arrivals during this slot, i.e., $\lambda_l(t)$ are assigned to channel-queues in proportion to the rates, i.e., $\lambda_l^c(t) = \dfrac{\lambda_l(t)r_l^c}{\displaystyle\sum_{b \in \mathcal{C}} r_l^b}$

Theorem 13. *The RPMMC scheduler stabilizes the queues in the network for any load-vector within the δ-reduced rate-region, where:*

$$\delta = \frac{\sigma_s}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}$$

Proof. The proof of stability is based on a Lyapunov drift argument. We present a proof-sketch here. The full proof can be found in Appendix C.

We adopt the following convention: at the beginning of each time-slot, the scheduling decisions are taken, and transmissions occur. Then new arrivals occur at the end of the slot (thus new arrivals cannot be transmitted in the same slot).

Let the queue-length of the queue for link l and channel c at the start of time-slot t be denoted by $q_l^c(t)$. Let the rate-allocated to link l in slot t over channel c be denoted by $x_l^c(t)$. Since we are considering single-interface nodes, at most one of the $x_l^c(t)$'s is non-zero for a link l. Furthermore $x_l^c(t) = 0$ if link l is not scheduled over channel c in slot t, and $x_l^c(t) = r_l^c$ else.

Also note that only link-channel pairs with $q_l^c(t) \geq r_l^c$ participate in the scheduling procedure during time-slot t.

Therefore, the queue dynamics are as follows:

$$q_l^c(t+1) = q_l^c(t) + \lambda_l^c(t) - x_l^c(t) \text{ where } \lambda_l^c(t) = \frac{\lambda_l(t)r_l^c}{\sum_{b \in \mathcal{C}} r_l^b} \tag{5.25}$$

We define the following Lyapunov function:

$$V_q(\overrightarrow{q}) = \sum_{l \in \mathcal{L}} \sum_{c \in \mathcal{C}} \left[\frac{q_l^c(t)}{r_l^c} \left(\sum_{k \in \mathcal{A}(l)} \sum_{d \in \mathcal{C}} \frac{q_k^d(t)}{r_k^d} + \sum_{k \in \mathbf{I}'(l)} \frac{q_k^c(t)}{r_k^c} \right) \right] \tag{5.26}$$

This Lyapunov function is somewhat similar in form to that used in [120]. It can be shown that this Lyapunov function satisfies the condition stated in Lemma 42 (Lemma 2 from [85]). This proves stability. For the detailed proof, please refer to Appendix C. □

Corollary 3. *When $\beta_s = 1$, the RPMMC scheduler's efficiency ratio is at least:*

$$\frac{|\mathcal{C}|}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}$$

112

Corollary 4. *The efficiency-ratio of the RPMMC scheduler is always at least:*

$$\left(\frac{\sigma_s}{|\mathcal{C}|}\right)\left(\frac{1}{K + \max\{1, \gamma\}}\right)$$

Proof. The proof follows from Theorem 13 and (5.1). □

5.8   On Scheduling with Heterogeneous Interfaces

The results presented in this chapter pertain to scenarios where the channels have heterogeneous characteristics, but the interfaces are all identical. Thus, it is of interest to consider how maximal scheduling algorithms may need to adapt in the face of heterogeneous interfaces, each of which may have constrained switching ability, and may only be able to operate on some subset of channels. The key distinction lies in the need to treat each node-link (pair of nodes capable of direct communication) as a set of distinct *radio-links* (corresponding to pairs of interfaces that could be used for communication). If a maximal schedule is computed in a manner oblivious to the interface heterogeneity, this can lead to performance degradation. We illustrate this via a very simple example:

Consider two mutually-interfering directed links $A \rightarrow B$ and $C \rightarrow D$. There are two channels 1 and 2 that both support the same data-rate r over both links. Node A has two radios, while all other nodes have one radio each. Nodes A and C both generate traffic at a constant rate $r - \epsilon$ (where ϵ is a very small positive constant). It is easy to see that $\gamma = 0, K = 1, \sigma_s = 2$ for this network. Hence, if all radios were identical and could operate on both channels 1 and 2, one would expect any maximal scheduler to achieve an efficiency ratio of 1 in this network.

However, in the considered scenario, the radios are heterogeneous, and many of them have constrained switching ability. The channel-sets on which these radios can operate are depicted in Fig. 5.3. The optimal scheduling decision in this scenario is to operate link $A \rightarrow B$ on channel 1 and link $C \rightarrow D$ on channel 2. A sub-optimal scheduler may schedule $A \rightarrow B$ on channel 2, thereby making it impossible to schedule $C \rightarrow D$ simultaneously.

Note that this latter schedule is a valid maximal schedule. However, it is computed in a manner oblivious to interface heterogeneity, and consequently, can lead to a very substantial performance degradation.
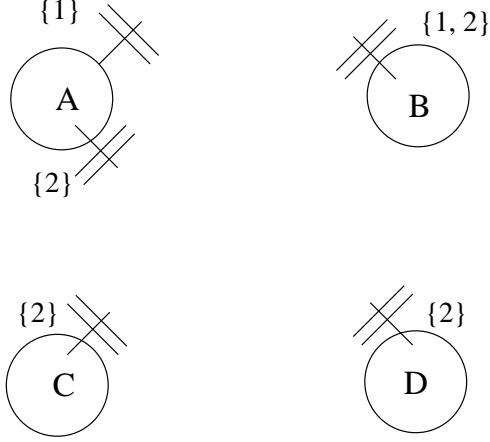
Figure 5.3: Example illustrating drawbacks of oblivious interface-selection

This motivates the importance of incorporating awareness of interface switching constraints into the scheduling algorithm.

We remark that it is indeed possible to adapt the algorithm of Lin-Rasool [74] to address heterogeneous radios (see [8]). We expect that it should be possible to similarly adapt the RPMMC scheduler to heterogeneous radios. This would be an interesting direction for future work.

5.9   Discussion

The intuition behind the RPMMC scheduler is very simple. By splitting the traffic across channels in proportion to the channel-rates, each link basically sees the average of all channel-rates as its *effective rate*. This helps avoid worst-case scenarios where the link may end up being repeatedly scheduled on a channel that yields poor rate on that link. Though exceedingly simple, the algorithm is made attractive by the fact that no information about queues at interfering links is required. Furthermore we showed that the efficiency-ratio of the RPMMC scheduler is always at least $\left(\frac{\sigma_s}{|\mathcal{C}|}\right)\left(\frac{1}{K+\max\{1,\gamma\}}\right)$ (Corollary 4). Note that $1+\beta_s(|\mathcal{C}|-1) \leq \sigma_s \leq |\mathcal{C}|$. Thus, the efficiency ratio of this algorithm does not degrade indefinitely as β_s becomes smaller.

5.10 Future Directions

The RPMMC scheduler provides motivation for further study of schedulers that work with limited information. The scheduler of Lin-Rasool and the RPMMC scheduler represent two extremes of a range of possibilities, since the former uses information from all interfering links, while the latter uses no such information. Evidently, using more information can potentially allow for a better provable efficiency-ratio. However, the nature of the trade-off curve between these two extremities is not clear. For instance, an interesting question to ponder is the following: If interference extends up to M hops, but each link only has information upto $x < M$ hops, what provable bounds can be obtained? This would help quantify the extent of performance improvement achievable by increasing the information-exchange, and provide insights about suitable operating points for protocol design, since control overhead can be a concern in real-world network scenarios.

Another direction for future work consists in characterizing network topologies in which the performance of greedy maximal scheduling in a multi-channel network with one or few interfaces per node is close-to-optimal.

Chapter 6

Channel/Interface Management in a Heterogeneous Multi-Channel Multi-Radio Network

In this chapter, we describe a proof-of-concept protocol for channel and interface management in a heterogeneous multi-channel wireless network. Our objective has been to incorporate awareness of radio and channel heterogeneity as well as traffic-awareness into the channel and interface management procedure. We have sought to leverage the insights from our theoretical results discussed in previous chapters of this dissertation, as well as insights from prior theoretical work in the literature. While we have designed our protocol in the context of 802.11 networks, with certain assumptions on node configuration, many aspects of the design, and many of the algorithms used, have broader relevance for a wide range of networks with heterogeneous radios and/or channels.

We begin by discussing related work in Section 6.1. We then describe the general architectural principles of our approach in Section 6.2. In Section 6.3, we describe the network and node model. In Section 6.4, we provide examples of various kinds of network conflicts that may need to be addressed by a channel and interface management protocol, and then discuss the protocol design in detail in Section 6.5. We describe simulation results in Section 6.6. In Section 6.7 we discuss some observations based on the protocol evaluation, and conclude in Section 6.8 by discussing some directions for future work.

6.1   Related Work

Protocols and architectures for multi-channel networks can be broadly categorized into those intended for single-radio devices, and those intended for multi-radio devices. In the case of single-radio devices, the channel coordination problem can be quite complex whereas, with multi-radio devices, the coordination issues are made somewhat easier to address by the presence of many radios.

Many protocols have been proposed for channel-coordination amongst devices having a

single radio each. A useful taxonomy for these has been described in [84]. Some protocols assume that all nodes are synchronized and follow a common hopping sequence when not sending data. A pair of devices wishing to send data stop hopping after negotiating a data-transfer, and stay on a common channel till it is over. Then they again start hopping as per the common hopping schedule. Instances include CHMA [111]. The class of split-phase protocols comprises those that utilize a notion of a negotiation phase during which nodes converge to a common channel and decide what channels to tune to for a window of time in the future. Prominent amongst these is MMAC [106], which uses a notion of ATIM window (similar to IEEE 802.11 PSM) to negotiate channels. Many proposals fall into the category of multiple-rendezvous protocols, e.g., SSCH [4], McMAC [105], Dominion [88]. In these protocols, nodes follow channel-hopping schedules that allow them to converge with each other sufficiently often. Of these, Dominion also includes a multi-channel routing component.

An approach termed component based channel assignment is proposed in [114], wherein all interfaces lying on the routes of intersecting flows are assigned the same channel. This keeps channel switching to a minimum.

Recently, there has been much interest in protocols/architectures for multi-channel multi-radio networks. Examples of multi-channel multi-radio testbeds include the Net-X project [67, 18, 12], a testbed at UCSB [96], and the Quail Ridge Reserve Mesh Network, UC Davis. Of these, the Net-X testbed is relevant to our work, as we adopt the node configuration used in Net-X.

Many protocols have been proposed to incorporate traffic awareness in various queueing and scheduling decisions, both for single and multi-channel scenarios. Neighborhood RED [122] proposes a variant of the RED algorithm, whereby queues at nodes within two hops are also taken into account, and not just the local queue. Warrier et al. have proposed a cross-layer architecture that is based on recent theoretical work on cross-layer optimization [117] Traffic-aware channel assignment in LANs has been considered in [97]. For LANs with uncoordinated access points, it has been proposed in [82], that channel-hopping can help prevent worst-case scenarios, and provide good average case performance. A traffic-oblivious joint routing and scheduling scheme for mesh networks has been proposed in [116]. Route/schedule computation is centralized, and worst-case congestion is minimized.

The 802.11 standard provides multiple physical layer specifications, and NICs for these are readily available off-the-shelf. There has been some work addressing the use of these radios of different types. Draves et al [29] have considered the issue of routing in a multi-channel mutli-radio mesh network where nodes are equipped with one radio each of type 802.11a and 802.11g. However, they do not consider the problem of channel selection.

The use of heterogeneous interfaces to handle route breakages has been proposed in [127]. In this work, nodes are equipped with primary 802.11a interfaces and secondary 802.11b interfaces. TCP flows use a primary path comprising the 802.11a interfaces, which is discovered via a reactive routing protocol. A proactive routing protocol is run over the secondary interfaces. When a link-breakage is detected, the TCP traffic can be immediately re-routed over the secondary path while a new primary path is being discovered.

Joint channel assignment and routing in a heterogeneous multi-channel multi-radio wireless network has been considered in [118]. This work targets a situation very similar to what we have considered in this chapter, and is closest in scope to our work. It allows for both heterogeneity in the operational abilities of interfaces, as well as in supported channel data-rates. It handles both single-radio, and multi-radio devices. A joint channel-assignment and routing scheme (JCAR) is proposed. However, this work treats the route for each flow as a sequence of interfaces, and therefore does not consider the possibility of link-layer data-striping. Moreover, it seeks a solution where interfaces switch channels only over substantially long periods of time.

The channel diversity in a multi-channel network provides opportunity for not merely load-balance but opportunistic selection of the channel with better channel quality. Opportunistic channel selection has been considered in MAC protocols such as MOAR [52], DB-MCMAC [14] and OMC-MAC [130]. However the global routing implications of local opportunism in a multi-hop wireless network have not been studied. Optimal channel probing strategies for a single-user multi-channel system have been studied in [40, 17]. The considered systems typically comprise one transmitter, capable of operating on N channels, which must select one channel for transmission. Self-organization based on measurements is considered in [53], and their approach consists of using a Gibbs sampler. Channel quality and rate-aware routing was addressed in [23].

6.2 General Design/Architectural Principles

We begin by briefly describing the general design and architectural principles on which we have based our protocol for multi-channel multi-radio wireless networks.

A Route as a Sequence of Nodes   A node-link is a pair of neighboring nodes. A radio link is a pair of radios on neighboring nodes. Thus, a node-link comprises a set of radio-links, and with suitable link-layer strategies, one can exploit this diversity/multiplicity. We adopt an approach of single-path routing with link-layer data-striping. Thus, a path from source to destination is a single sequence of nodes (and hence also a series of node-links). When packets need to be transmitted over a node-link, the link layer determines which radio(s) and channel(s) to use. Thus, the link-layer can perform link-level data-striping if many radios are available at both transmitter and receiver. Moreover, when there are multiple flows that pose interference or/and interface conflicts for each other, this approach allows flexibility in adapting on the fly, as the link layer can make packet scheduling decisions at fine granularity.

Channel Restriction   While one would like to exploit the available channel diversity to improve throughput, doing so effectively would require some mechanism to sample/probe channels, as well as exchange of information about channel state/quality. This cost can be significant, especially if the number of available channels is large. Moreover, in a distributed setting, when multiple entities act independently, opportunism can have an adverse effect on load-balance, e.g., consider a worst-case scenario where all nodes in a vicinity decide that channel x has best quality and start using that channel simultaneously.

One would typically expect that much of the benefit of opportunistic exploitation of channel diversity can be obtained by having the choice of a few channels, and thus a reasonable solution lies in restricting the operation of a link to a subset of all possible channels available to it (a *channel pool*). One can then attempt to opportunistically exploit diversity amongst channels in this *channel pool*. We note that some prior work, e.g., [115], has studied this issue in a single-hop setting and concluded that a few channels indeed provide a good trade-off between diversity-gain and probing cost. The same conclusion is likely to hold even in multi-hop settings.

Moreover, channel-restriction has the potential to provide a degree of a priori load-balance (since different links will have different channel pools). This can help reduce the possibility of worst-case channel-selection scenarios link the one mentioned above, while still providing enough choices to each link for good load-balance. Some intuition for this can be derived from our result for random (c, f) assignment described in Chapter 4, as well as past work on balls and bins with choices [3, 83].

We propose the following simple channel restriction policy: each interface is assigned a small pool of f channels for substantial periods of time. The channel pools are chosen and adjusted so that, within the two-hop neighborhood of any interface, each channel occurs in the pool of approximately the same number of interfaces.

The current channel for each interface is selected more frequently.

It is to be noted that *the poolsize f provides a control knob to tune the degree of dynamism of the protocol*. Setting $f = 1$ corresponds to a largely static channel assignment (where interfaces switch channels very infrequently), while setting $f = c$ corresponds to a fully dynamic assignment, in which the current channel may be chosen from the entire set of possible channels.

Late Binding of Packets to Channel/Interface Since we intend to perform dynamic channel selection over intermediate time-scales, it is beneficial to defer the binding of an outgoing packet to a channel and interface to as late a stage as possible without significantly affecting efficiency. This allows for greater flexibility and adaptivity.

Channel Cost Formulation Incorporating Awareness of Traffic Levels and Conflicts Two kinds of conflicts can limit performance in a multi-channel network:

1. *Interference Conflicts:* A channel becomes the bottleneck due to traffic overload

2. *Interface Conflicts:* A radio-interface becomes the bottleneck due to an overload of traffic it is expected to relay.

Thus, a link cost metric for scheduling should try to capture these two conflicts, so that channel/interface selection decisions are able to address them effectively.

Use of limited information from vicinity A wireless transmission can create interference for other transmissions over a distance corresponding to many hops, depending on the
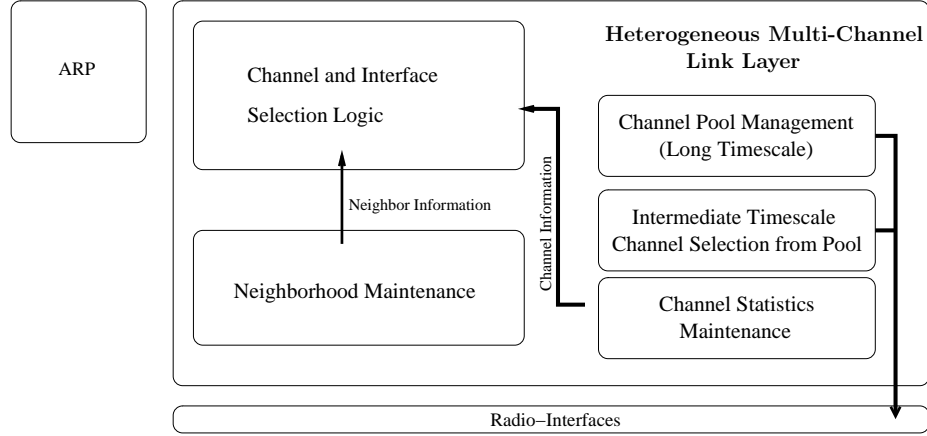
Figure 6.1: General Architectural Template

transmission powers, rates, and corresponding SINR requirements. Moreover, the choice of carrier-sense threshold also affects the degree of spatial reuse achievable. If the carrier-sense threshold is conservatively set to a large value, a single ongoing transmission can block other transmissions over a large area extending well beyond its two-hop neighborhood. If the region over which a link can potentially create conflict extends over K hops, where K is large, then it may not be feasible to provide a node information about this whole region due to concerns about high overhead, as well as large delays, because of which the information may become stale by the time it is received. Thus, it is desirable to operate using limited exchange of explicit information, and use implicit feedback mechanisms to infer network and channel conditions. Therefore, in the proposed design approach, nodes only have explicit information up to two hops, but use contention on a channel as an implicit indicator of traffic levels.

A high-level schematic of the envisioned framework incorporating the elements described above is depicted in Fig. 6.1.

6.3 The Model

We assume a node configuration similar to the Net-X Project [64] where interfaces are classified as belonging to one of the following two categories:

1. *R-interface:* A R-interface is used for receiving packets, and whenever its channel is changed, the change is advertised to neighbors. A R-interface is also used for

transmitting packets that are to be sent on its current channel.

2. *T-interface:* A T-interface is used for transmitting packets. When a packet is to be transmitted to a next-hop node, a T-interface is switched to one of the R-channels of the next-hop node, and used to transmit the packet.

The interfaces can be of type: single-mode 802.11a, single-mode 802.11g and multi-mode 802.11ag.[1]

Each node is assumed to either have at least one R-interface and one T-interface of type x or no interface of type x, where x can be 802.11a or 802.11g. A multi-mode 802.11ag radio can be present as a T-interface, and can be counted towards each type, e.g., if a node has one R-interface each of type 802.11a and 802.11g, and a T-interface of type 802.11ag, then this is a valid configuration. Currently, we do not allow multi-mode R-interfaces.

Note that the above classification into R-interfaces and T-interfaces is purely a link-layer characteristic, based on how the link layer intends to utilize each interface; each interface of a particular type is otherwise identical, and has the same physical and MAC layer properties.

Adopting this dual-radio framework helps avoid connectivity issues, and channel coordination problems such as multi-channel *deafness* [79], and enables us to focus on the scheduling aspects of the problem.

At each node, we have a single link-layer entity that manages all interfaces (which perform independent MAC procedures). Since we wish to perform single-path routing while allowing for the possibility of transparent link-layer striping, we require all interfaces of a node to have the same IP address. To avoid changing ARP, all interfaces of a node are also assigned the same MAC address.

Interfaces are assumed to be capable of fairly fast switching. More specifically, we consider that switching between channels in the same mode takes $250\mu s$ (this is consistent with channel switching times reported in recent work, e.g., [41]). If a mode-switch is also required while doing the channel switch, then we assume that the time taken is $500\mu s$, since a mode-switch might typically take more time than a simple channel-switch.

We have designed a channel and interface management protocol for this described model. For evaluation with multi-hop flows, we use manually specified routes, wherever needed.

---

[1]As we mention later, 802.11b is not considered separately, as we currently fix the 802.11b/g rate at 2 Mbps, and thus the two are effectively the same, if 802.11g is operated in backward compatibility mode (which is what we assume).

Currently, we do not consider dynamic rate adaptation. The data-rate for all 802.11a communication is 6Mbps, while that for all 802.11g communication is 2 Mbps. We do remark that the link layer algorithms can operate in the presence of a rate-adaptation algorithm, with suitable link-rate feedback from the MAC. However, our current goal is to study the channel and interface management aspects without regard to interaction with rate-adaptation. Incorporating a suitable auto-rate fallback algorithm at the MAC, and providing appropriate rate feedback to the link layer, would be an interesting direction for future work.

The RTS/CTS mechanism is effectively disabled in the 802.11 MAC protocol by choosing a very high value for *RTS_Threshold*. Physical carrier sense is used. 802.11g uses 2 Mbps as the data rate for all packets (including broadcast and ACK packets). The PLCP datarate is 1 Mbps for 802.11g, while it is 6Mbps for 802.11a. 802.11g operates in backward compatibility or mixed-mode and uses the same MAC parameters as 802.11b.

Since the link layer may perform data-striping over a link, there is a possibility of out-of-order packet delivery, and thus reordering of packets may be required. Currently, we do not address this issue, as reordering can also be done at the receiving transport endpoint. However, we discuss the issue of implementing a reordering buffer at the link-layer in Section 6.8.

6.4 Interference and Interface Conflicts

As was discussed in Section 6.2, the channel cost metric should be able to capture both interference and interface conflicts. Before we move on to describe our protocol, and how it addresses this issue, let us consider a few illustrative examples in the context of the specific network and node model we are considering. In these examples, each node has one 802.11a R-interface and one 802.11a T-interface, and for the purpose of simplicity, we assume that ideal TDMA scheduling is possible. The transmission rate in use is 6 Mbps.

Example 1. *Consider the situation in Fig. 6.2. There are only two 802.11a channels available for use (let us denote them by 1 and 2). All links interfere with each other.*

Consider two different traffic patterns:

1. Link l_1 has traffic-demand 6 Mbps, while links l_2 and l_3 have traffic demand 3 Mbps
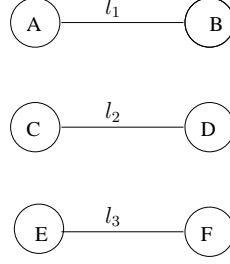
Figure 6.2: Example 1: Interference Conflicts

each. An ideal scheduler can meet these demands by having l_1 operate on channel 1 and l_2 and l_3 operate on channel 2. A traffic-unaware static distributed channel assignment strategy's best solution is to have two of these links on one channel, in a manner oblivious to actual load. Thus, it could potentially operate l_1 and l_2 on channel 1 and l_3 on channel 2, resulting in throughput degradation.

2. *Each link l_1, l_2, l_3 has a single-flow with traffic-demand 4 Mbps. An ideal scheduler can have links l_1 and l_2 operate over channels 1 and 2 respectively, and have l_3 time-share between channels 1 and 2, as follows: in a unit interval $[0 : 1]$ the following schedule is followed: $[0 : \frac{1}{3}] : l_1$ transmits over channel 1, l_3 transmits over channel 2; $[\frac{1}{3} : \frac{2}{3}] : l_1$ transmits over channel 1, l_2 transmits over channel 2; $[\frac{2}{3} : 1] : l_3$ transmits over channel 1, l_2 transmits over channel 2. This allows all traffic demands to be met. A static and traffic-unaware channel-assignment strategy would not be able to achieve this.*

Now consider an example illustrating a potential interface conflict and how it can be resolved:

Example 2. *Consider the situation in Fig. 6.3. There are 3 802.11a channels available for use. There are two flows: $X \to Y$ and $X \to Z$ with traffic demand 6 Mbps each. If the R-interfaces of all 3 nodes are on different channels, the maximum aggregate throughput possible is 6 Mbps. However, if the R-interface of either Y or Z is on the same channel as the R-interface of X, while the R-interface of the remaining node is on another channel, then both flows can get 6 Mbps, since X can use its R-interface to transmit packets to one, and its T-interface to transmit packets to the other. A traffic-unaware strategy that only considers interference conflicts in a combinatorial sense (number of interfering interfaces*
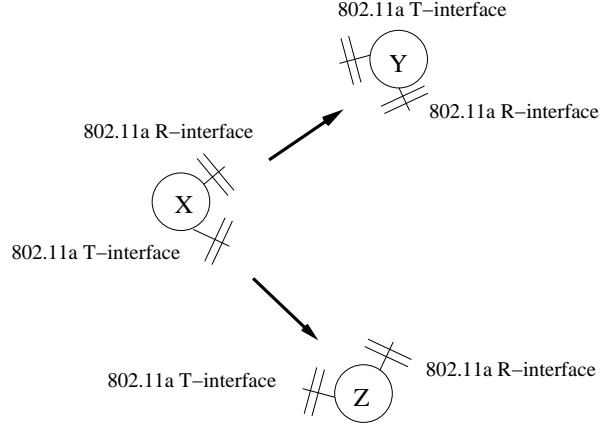
Figure 6.3: Example 2: Interface Conflicts

on a channel) would not be adequate for this; in fact, such a strategy would typically try to assign different channels to all 3 R-interfaces.

6.5 The Heterogeneous Multi-Channel Link Layer (HMCLL) Protocol

The proposed link layer protocol, which we term the Heterogeneous Multi-Channel Link Layer (HMCLL) Protocol, can be said to lie in Layer 2.5, i.e., between layers 2 and 3 in the protocol stack. The HMCLL is IP-aware. This IP-awareness has two benefits:

- HMCLL control packets have IP headers, and the HMCLL can cache IP-to-MAC mappings in the ARP table. This provides resilience to issues caused by ARP losses (see [15] for an exposition on ARP-loss related problems in wireless networks).

- The HMCLL can provide the network layer with a cost associated with a link to a next-hop node, identified by the network layer via its IP address. While the focus of the current work is on designing an intelligent link layer protocol, it is of great interest to consider future work where the link-layer provides an abstracted cost metric to a routing protocol. We discuss future directions in Section 6.8.

The HMCLL protocol aims to handle scenarios with different number and type of interfaces, and channels with different rates. Many of the HMCLL algorithms are conceptually formulated in fairly general terms where *each channel is characterized by the rates achievable on*

different links using that channel, each interface is characterized by the set of channels on which it can operate, and the relationship between channels is characterized by the extent to which they compete for interface-time at a node [2]. Thus, they could be applied to a wider range of radio-types, provided an appropriate characterization of the above elements is made available to them.

Note that different channel rates may arise due to various reasons, e.g., (a) as a result of different modulations providing different transmission rates (e.g. we use 6 Mbps for 802.11a and 2 Mbps for 802.11g), or (b) as a result of variable channel quality leading to different packet loss rate (and hence different net rate). While most of the protocol algorithms are oblivious of the reason for the different rates (and just use information about achievable rates for making decisions), we do remark that there is an important practical distinction one must be aware of: rate differences due to different modulations are known accurately a priori, whereas rate differences due to variable channel quality require good channel estimation techniques to determine with fair accuracy. While most algorithms used by the protocol are applicable in either scenario, achieving good performance in environments with highly dynamic channel conditions will require that good estimates of achievable rates be available, which in turn would require improved channel-estimation techniques beyond the rudimentary estimation mechanisms used by the current design. Similarly, the current simplistic neighborhood management would need much improvement. We discuss this issue further in Section 6.8.

6.5.1   Neighborhood and Channel/Traffic Statistics Maintenance

We begin by introducing some terminology. The one-hop neighborhood of a node u is denoted by $nbd(u)$, and its two-hop neighborhood is denoted by $nbd_2(u)$. In this chapter, u is not considered to be included in $nbd(u)$ or $nbd_2(u)$.

Each node u has a set of active interfaces $\mathcal{M}(u) = M_R(u) \cup M_T(u)$, where $M_R(u)$ and $M_T(u)$ are the R-interfaces and T-interfaces respectively of node u. Let $\mathcal{C}(x)$ denote the set of channels on which interface x is capable of operating. Each interface has a type denoted by $type(x)$ which uniquely determines the set of channels $\mathcal{C}(x)$ on which x can operate[3].

---

[2]There exist other aspects to the relationship between channels, e.g., adjacent channel interference, and one could potentially try to extend the characterization to include these. However, that is beyond the scope of the current work, which assumes orthogonal channels

[3]For instance, we currently consider three types: 802.11a, 802.11g, and 802.11ag. Of these, only 802.11a

Each R-interface x has an associated subset of channels called the channel-pool $\mathcal{P}(x) \subseteq \mathcal{C}(x)$ such that $|\mathcal{P}(x)| = f$. The current channel of interface x is denoted by $c(x)$. We use the notation $c(\mathcal{S})$ where $\mathcal{S}$ is a set to denote $\bigcup_{x \in \mathcal{S}} \{c(x)\}$. An interface is said to be active if it is in use (i.e., has not been deactivated by the LL).[4]

The link layer maintains the following information:

- *A List of One Hop Neighbors:* This contains an entry for each node in $nbd(u)$ known to u. Each neighbor entry has a *LifeTime* field, as well as *aLifeTime* and *bLifeTime* fields. It is also marked as symmetric or asymmetric. If the LL receives a new packet from the higher layers with a next hop node that is currently marked asymmetric, it drops the packet. Each entry also has a reachability flag for each of 802.11a and 802.11g based on the respective lifetime value; these a/g-specific attributes are maintained primarily to provide a basic binary measure of achievable rate (0 or the raw data-rate) in the absence of any accumulated rate history.

- *A List of all 2-hop Neighbors:* This contains an entry for each node in $nbd_2(u)$ known to u.

- *Statistics about each local interface:* An estimate of interface TX-utilization for interface x, denoted by $\rho(x)$, i.e., the fraction of time the interface was busy doing work related to transmitting (contending, transmitting, switching) is computed. Utilization is computed over intervals of duration $T_{rassign}$, and an average utilization estimate is maintained as an EWMA updated as $\overline{\rho}(x) = 0.25 * \overline{\rho}(x) + 0.75 * \rho(x)$.

- *The following statistics are maintained about each channel on which some local interface can operate:*

    - Effective Transmission Rate for a link, denoted by $r(u, v, c)$: For each packet sent by u to v over channel c, the MAC provides the LL feedback on the number of transmission attempts needed ($x(u, v, c)$), as well as the raw datarate used (R). The success rate $\psi(u, v, c)$ is maintained as an EWMA, and updated as follows:

$$\psi(u, v, c) = 0.25 * \psi(u, v, c) + 0.75 * \frac{1.0}{x(u, v, c)} \tag{6.1}$$

and 802.11g are valid types for R-interfaces.

[4]Some interfaces may be deactivated if the LL is unable to assign all local interfaces distinct channels, e.g., when the number of channels available for use is smaller than the number of interfaces at the node.

The instant effective rate is $r_{new}(u,v,c) = R * \frac{1.0}{x(u,v,c)}$. The LL maintains $r(u,v,c)$ as an EWMA, which is updated as follows:

$$r(u,v,c) = 0.25 * r(u,v,c) + 0.75 * r_{new}(u,v,c) \qquad (6.2)$$

If the last update of $r(u,v,c)$ occurred more than $2 * T_{QINFO}$ time ago, $\psi(u,v,c)$ is reset to 1 and $r(u,v,c)$ is reset to $NO\_RATE\_HISTORY$.

- Net Data Rate for a link, denoted by $\mu(u,v,c)$: this is the net time taken to transmit a packet, when taking into account the time spent in contention, i.e., backoff, etc, as well as any retransmissions. This is maintained as an EWMA. Whenever the LL gets feedback from the MAC that the total time taken in transmitting a packet was μ_{new}, it updates the estimate as $\mu(u,v,c) = 0.9 * \mu(u,v,c)_{old} + 0.1 * \mu_{new}$. If the last update of $\mu(u,v,c)$ occurred more than $2 * T_{QINFO}$ time ago, $mu(u,v,c)$ is reset to $NO\_RATE\_HISTORY$.

- Average Contention Time experienced by u when transmitting a packet on channel c, denoted by, $\kappa(u,c)$. This is also maintained as an EWMA. Whenever the LL gets feedback from the MAC that a packet required contention time k on channel c, we use the following update equation: $\kappa = 0.9 * \kappa + 0.1 * k$.

Note that all rate estimates above are in units of bits per second.

Neighborhood management, as well as channel and traffic statistics maintenance are facilitated by exchange of link layer control packets.

For each $v \in nbd(u)$, u maintains a set $\mathcal{T}(u,v) \subseteq M_R(v)$, which is the set of R-interfaces of v that u would be willing to send packets to. The choice of $\mathcal{T}(u,v)$ can be used to allow/disallow link-layer data-striping (e.g., if $|M_R(v)| > 1$ but $|\mathcal{T}(u,v)| = 1$, then this corresponds to no data striping). Currently, we use $\mathcal{T}(u,v) = M_R(v)$. However, in the rest of the description, we will continue to use the term $\mathcal{T}(u,v)$ to highlight that the link layer algorithms can work for other choices of $\mathcal{T}(u,v)$ (of course, in that case, an additional algorithm will be needed to select $\mathcal{T}(u,v)$).

The link layer also maintains a system of queues (described later in this chapter). These include a queue of outgoing packets to each next-hop neighbor. The length of the queue (in bits) for neighbor v at node u is denoted by $q_{nbr}(u,v)$. There is also a queue for each

channel. The length of the queue (in packets) for channel c is denoted by $q_{ch}^{p}(u, c)$.

We also use the following definitions and notation:

The minimum-rate constant θ is a small constant chosen such that θ is much smaller than the typical values of achievable rates. The primary purpose of θ is to avoid division-by-zero anomalies when computing various quantities of interest. In the current design, we use $\theta = 1$ (as the typical rate values are of the order of 10^6 in bits/sec).

The ratesum for a link (u, v) is denoted by $\sigma(u, v)$ and defined as:

$$\sigma(u, v) = \sum_{y \in \mathcal{T}(u,v)} r(u, v, c(y))$$

Intuitively, the significance of the ratesum is that the LL needs to estimate the load on each channel in the near future. To do so, it pretends that each neighbor v splits traffic it sends to u across channels in $\mathcal{T}(u, v)$ in proportion to the channel-rates, and therefore, the ratesum plays a role in computing various estimates, as will be evident (v may not necessarily split traffic in this manner, but it serves as a reasonable hint for LL decisions). Note that this is reminiscent of the RPMMC scheduler described in Chapter 5, from which we drew intuition for this approach.

The link-layer at u tracks the number of bits sent to v over intervals of duration $T_{rassign}$, denoted by $s(u, v)$. Average sent bits for link (u, v) are denoted by $\bar{s}(u, v)$, and maintained as an EWMA. At the end of every period, $\bar{s}(u, v)$ is updated as:

$$\bar{s}(u, v) = 0.25 * \bar{s}(u, v) + 0.75 * s(u, v)$$

Interface-conflict cost for channel c over link (u, v) is defined as follows (in the following text K is a suitably chosen threshold constant):

1. If $q_{nbr}(u, v) < K$ then $\chi(u, v, c) = 0$

2. If $q_{nbr}(u, v) >= K$:

   (a) If c is an R-channel of u, i.e., there is $x \in M_R(u)$ such that $c(x) = c$, then it is

defined as:

$$\chi(u,v,c)$$

$$= \sum_{w\in nbd(u)} \left(\frac{q_{nbr}(u,v)}{\max\{\sigma(u,v),\theta\}} + T_{rassign}(\rho(x)-0.8)_+ \right) I_{(\exists\ y\in\mathcal{T}(u,w):c(y)=c)}$$

(b) If c is not an R-channel, let $\mathcal{S}(b) \subseteq M_T(u)$, be the set of T-interfaces of u that can operate on a channel b. Then:

$$\chi(u,v,c) = h(u,v,c) + U(u,c)I_{h(u,v,c)>H}$$

where

$$h(u,v,c) = \frac{1}{|\mathcal{S}(c)|} \sum_{x\in\mathcal{S}(c)} \left(\sum_{w\in nbd(u)} \sum_{\substack{y\in\mathcal{T}(u,w) \\ c(y)\notin c(M_R(u)) \\ \wedge c(y)\in\mathcal{C}(x)}} \frac{q_{nbr}(u,v)}{\max\{\sigma(u,v),\theta\}|\mathcal{S}(c(y))|} \right)$$

$$U(u,c) = \frac{T_{rassign}}{|\mathcal{S}(c)|} \sum_{x\in\mathcal{S}} (\rho(x)-0.8)_+ \text{ and } H \text{ is a suitably chosen threshold}$$

To provide some intuition for the relevance of this quantity, it provides an estimated measure of the amount of traffic (normalized by rate) that contends for interface time at sending neighbor v on the interface(s) that are used to send packets on channel c. The utilization-based component is included primarily because when we have TCP traffic, the queues may never become large enough to trigger a change in channel assignment; in those scenarios tracking interface utilization becomes important, as a heavily utilized interface implies a large conflict cost.

The local interface conflict seen by channel c at node u is denoted by $\chi_{local}(u,c)$ and defined as:

1. If c is the current channel of a local R-interface, $\chi_{local}(u,c) = 0$.

2. If c is not an R-channel:

$$\chi_{local}(u,c) = \frac{1}{|\mathcal{S}(c)|} \sum_{x\in\mathcal{S}(c)} \sum_{\substack{d\in\mathcal{C}(x) \\ d\neq c\wedge d\notin c(M_R(u))}} \left\lceil \frac{q_{ch}^p(u,d)}{|\mathcal{S}(d)|} \right\rceil \tag{6.3}$$

where $\mathcal{S}(b)$ denotes the set of T-interfaces at the local node u that can operate on

130

Name	Description
T_{LLINFO}	Used to determine interval between consecutive LLINFOs
J_{LLINFO}	Used to determine random jitter between consecutive LLINFOs
T_{QINFO}	Used to determine interval between consecutive QINFOs
J_{QINFO}	Used to determine random jitter between consecutive QINFOs
T_{pool}	Interval between invocations of Channel Pool Management Algorithm
$T_{rassign}$	Base interval between execution of R-channel selection algorithm at an interface (a random jitter gets added to it)
NBR_TTL	Maximum Time-To-Live of a neighbor entry
IFR_TTL	Maximum Time-To-Live of a 2-hop neighbor entry
K	Threshold value used in computing χ (unit is bits)
H	Threshold value used in computing χ (unit is seconds)
$\delta_{inertia}$	Minimum difference in channel cost required for new R-channel selection Used to provide hysteresis in R-channel selection decision; $\delta_{inertia} > 0$
δ_{min}	Used to provide hysteresis in R-channel selection decision
δ_{comb}	Used to determine whether R-channel selection should use combinatorial criteria

Table 6.1: Protocol Parameters

channel b.

The intuition behind $\chi_{local}(u, c)$ is that it provides a quantification of the conflict faced by packets bound to channel c from packets bound to channels that compete with c for local interfaces.

Total incoming data score for interface $x \in M_R(u)$ with respect to channel b is defined as:

$$Incoming(x, b) = \sum_{v \in nbd(u)} \left(\frac{s(v, u) + q_{nbr}(v, u)}{\max\{[\sigma(v, u) - r(v, u, c(x)) + r(v, u, b)], \theta\}} \right)$$

Incoming queue score for an R-interface x at node u is defined as:

$$\eta(x) = \sum_{v \in nbd(u)} \frac{q_{nbr}(v, u)}{\max\{\sigma(v, u), \theta\}}$$

$\eta(x)$ provides an estimated measure of the amount of traffic queued at neighbors of u that is expected be sent to interface x.

For clarity, various parameters used by the LL are tabulated in Table 6.5.1.

Link Layer Control Packets

The link layer sends/receives the following control packets:

1. **LLINFO:** This packet is broadcast by each node u. Thus a copy is sent on each channel c such that some interface of u can transmit on c. The LLINFO is sent after intervals of duration $MAX(T_{LLINFO}, (0.15 * m)) + X$, where m is the total number of channels available to the network (on which copies of the LLINFO may possibly need to be sent), and X is a random variable uniformly distributed in $[0, J_{LLINFO}]$. It may also be triggered by events that require fresh information propagation (e.g., a change of an R-interface's channel, or pool membership). The contents of an $LLINFO(u)$ packet are as follows:

   - Sequence number

   - Number of active R-interfaces

   - For each active R-interface $x \in M_R(u)$:
   $ID(x), type(x), |\mathcal{P}(x)|, c(x), \{b|b \in \mathcal{P}(x)\}, \eta(x)$

   - For each $v \in nbd(u)$:
   $seqno, \forall y \in M_R(v) : \{ID(y), type(y), |\mathcal{P}(y)|, c(y), \{b|b \in \mathcal{P}(y)\}, \eta(y)\}$

   Though in our current simulator implementation, we use a globally unique $ID(x)$ for each interface x, we remark that one only requires that each node maintain a locally-unique ID for each of its interfaces, since the pair *(nodeIP, ID)* then provides a globally unique identification for each interface.

2. **QINFO:** A QINFO$(u \rightarrow v)$ packet is unicast by each node u to some or all neighbors in situations where the number of channels is greater than 1 and the poolsize is also greater than 1. The QINFO sending routine is invoked after intervals of duration $T_{QINFO} + X$, where X is a random variable uniformly distributed in $[0, J_{QINFO}]$. To reduce overhead, if $|nbd(u)| < 5$, u sends a QINFO to each $v \in nbd(u)$ that is a symmetric neighbor, else it sends a QINFO to those symmetric neighbors v for which $q_{nbr}(u, v) + \bar{s}(u, v) > 5000$ (note that the unit is bits). This packet contains the following information:

   - Length of outgoing queue to neighbor: $q_{nbr}(u, v)$ and recently sent data $s(u, v)$

   - Number of active R-interfaces at v known to u (this will be $|M_R(v)|$ unless u has wrong information about v)

- For each R-interface $y \in M_R(v)$:

  $|\mathcal{P}(y)|,\ c(y),\ \forall\ c \in \mathcal{P}(y)$: $r(u,v,c), \kappa(u,c), \chi(u,v,c)$

3. **CINFO:** A CINFO$(u \to v)$ is sent by u to $v \in nbd(u)$ if u receives a QINFO from neighbor v containing incorrect information about u's interfaces. The contents of a $CINFO(u)$ packet are as follows:

   - Sequence number

   - Number of R-interfaces of u

   - For each R-interface $x \in M_R(u)$: $ID(x), type(x), |\mathcal{P}(x)|, c(x), \{b|b \in \mathcal{P}(x)\}, \eta(x)$

4. **PROBE:** A probe packet is a broadcast packet which is periodically sent with the sole purpose of estimating contention on each channel. This packet does not contain any information.

The sequence numbers for the LLINFO and CINFO packets are drawn from the same 32-bit sequence number space, and the sequence number is incremented after each packet is sent. QINFO and PROBE packets have no sequence number.

The link layer at node u updates its local information on receipt of control packets in the manner described below:

LLINFO: Whenever an LLINFO is received from v, if v is not already in the neighbor-list, a new entry is created. The *LifeTime* field of the (new or pre-existing) neighbor entry is set to *NBR_TTL*. If an LLINFO is received by u from v on an 802.11a channel, it marks v as reachable using 802.11a, and sets the *aLifeTime* field as *NBR_TTL*. Similarly, if an LLINFO is received on an 802.11b/g channel, it marks v as reachable using 802.11b, and sets the *bLifeTime* field as *NBR_TTL*. The *aLifeTime* and *bLifeTime* fields are refreshed whenever LLINFO packets are received on the appropriate channels. A periodic timer checks for expired entries. If an entry expires, the corresponding reachability flag is set to false. In the absence of any other feedback, this reachability information is used to determine the achievable rate from u to v on a channel c. We remark that this approach is flawed in that u receiving a packet from v indicates that u is reachable from v and not that v is reachable from u. Thus, this approach basically inverts the reachability information. However, it provides a low-overhead way to ensure that unless both nodes receive packets from each

other, the link will be marked as *asymmetric*, and the LL will not accept any new packets from higher layers to send to this neighbor.[5]

If the sequence number on this packet is not smaller than or equal to the last sequence number received from v, the interface and pool-channel information is overwritten, and the neighbor information is also processed, else the packet is discarded after refreshing lifetime and reachability information.

When an LLINFO is received from some neighbor, containing a record for v as 2-hop neighbor, if v is not already in interferer-list, a new entry is created. The lifetime of the (new or pre-existing) interferer entry is set to MAX_IFR_TTL. If v is also an existing 1-hop neighbor, and the sequence number on this entry is not smaller than or equal to the last sequence number associated with v's entry, the interface and pool-channel information is overwritten. For 2-hop neighbor entries, it is always overwritten (this can be extended to perform the sequence number check on existing 2-hop neighbors too).

If the received LLINFO leads to a change in important information about the neighbor's interfaces (i.e., number of R-interfaces, or current channel of an R-interface), a new LLINFO is sent out to propagate the changed information to other neighbors. The sending of a fresh QINFO to this neighbor may also be triggered. Moreover, if the LLINFO indicates that an R-interface of a neighbor v has changed its channel from c_{old} to c_{new}, any packets with next-hop v that are enqueued in $Q_{ch}(u, c_{old})$ are flushed.

QINFO: Whenever a QINFO is received from a neighbor v, if v is not in u's neighbor-list, no action is taken. If v is indeed in the neighbor-list, information in QINFO overwrites all information received from previous QINFO packets. Also, depending on whether it was received over an 802.11a channel or an 802.11b/g channel, the *aLifeTime* or *bLifeTime* field is reset to NBR_TTL, and the corresponding reachability flag is also set. The incoming queue information stored from a QINFO expires after a certain interval (the LL runs a timer that periodically checks when the last QINFO was received from a neighbor. If the time elapsed since the last QINFO is greater than $3 * T_{QINFO}$, the information about $q_{nbr}(v, u)$ and $s(u, v)$ is reset to 0).

---

[5]In highly dynamic situations, where the status of a neighbor may fluctuate between symmetric and asymmetric, this can lead to an incorrect view and resultant loss of performance. It can be improved upon by including information in the LLINFO packet as to whether packets were received from a neighbor on 802.11a and/or 802.11g in the recent past, and using the information received about oneself from one's neighbor to assess directional reachability and determine the default achievable rate.

CINFO: If a CINFO is received from v with a fresher sequence number than the last one received from v, the interface and channel pool information in the CINFO overwrites prior information. A CINFO receipt can also be used to assess a/b-reachability.

6.5.2 Interface Management

As has been described earlier, interfaces are classified as being either R-interfaces, or T-interfaces.

Since all interfaces at a node are assigned the same MAC address, but have independent MAC procedures, it is important to take care that at any time instant, if some R-interface of node u is tuned to a channel c, then no other R-interface or T-interface of u should be tuned to c at that time. Otherwise, the following undesirable scenario may possibly occur: suppose neighbor v is sending data to u on channel c. Since u has two interfaces tuned to channel c, and both have the same MAC address, they will both receive the packets, and believe that they are the intended recipients. Thus, they will both send ACKs. As a result, the ACKs may collide, in which case, v would consider the packet lost, and retransmit. Repetition of the same could lead to throughput degradation. The HMCLL protocol tries to avoid the possibility of an R-interface and another interface being tuned to the same channel simultaneously, except for rare and brief transient periods that may arise when one or more interfaces are switching. While there may potentially be occasional periods when more than one T-interfaces are on the same channel, this does not cause the wasteful transmission problem due to multiple ACKs, as packets intended for a node are sent only on the channel of an R-interface. If two T-interfaces happen to each be on the same channel, physical carrier sense addresses the issue that only one of them should transmit at a time. Thus, while such a scenario may sometimes lead to a waste of interface time (if there are packets waiting to be sent on another channel that the interface can operate on), this does not cause any serious issues.

Except for link layer control packets, packets received on a T-interface are discarded by the LL, to avoid the possibility of receiving duplicate packets (primarily true for broadcast packets). However link layer control packets are processed in the same way as packets received on an R-interface. This helps provide resilience to loss of control packets sent on the R-interface's channel. It does not affect correctness as the operations performed

on receipt of a control packet are idempotent (new information in a packet completely overwrites previous information). The possibility of a delayed control packet being received and causing stale information to overwrite newer information is made negligible by using sequence numbers for the control data sent by any single neighbor. and ignoring packets with a sequence number smaller than or equal to the last known sequence number (where smaller is defined as in [31]).

R-Interface Management

Following the channel restriction approach we described in Section 6.2, we associate with each interface a pool of channels, from which the current channel is dynamically selected. Thus, the R-interface management has two aspects, viz., channel pool management, and R-channel selection. We now describe each of these.

Channel Pool Management Recall that $\mathcal{C}(x)$ denotes the set of channels on which interface x is capable of operating, each R-interface x has an associated channel-pool $\mathcal{P}(x) \subseteq \mathcal{C}(x)$ such that $|\mathcal{P}(x)| = f$, and the current channel of an interface x is denoted by $c(x)$. Note that one could potentially allow different pool sizes for different interfaces, but for simplicity, this is currently a global constant for all interfaces of a particular type.

In keeping with the objective of a priori load-balance, it is desirable that the channels be equitably distributed across pools, such that in any vicinity all channels occur in roughly the same number of pools.

We use a probabilistic mechanism for pool management.

At the time of starting up, each interface is assigned a set of f channels chosen uniformly at random from all such possible f-subsets. Progressively, as LLINFO packets are received from neighboring nodes, the Neighbor Table gets populated with information about the channel-pools of the R-interfaces of these nodes. The Channel Pool Manager uses a timer that is scheduled at start-up after an interval uniformly distributed between 0 and T_{pool} seconds, and thereafter rescheduled every T_{pool} seconds. The initial random interval serves to desynchronize the pool-selection decisions of different nodes. Whenver the timer expires, the procedure described in Algorithm 1 is executed.

In the current design, the periodic channel pool management algorithms of all R-interfaces at node u use the same timer (i.e, they are all executed sequentially whenever

Algorithm 1 Channel Pool Management Algorithm (Interface x)

$I(x) \leftarrow$ the set of all R-interfaces within 2 hops of interface x
for all $c \in \mathcal{C}(x)$ **do**
    $n(c) \leftarrow |\{y|y \in I(x) \cup \{x\}, c \in \mathcal{P}(y)\}|$
end for
$\overline{n} \leftarrow \frac{1}{|\mathcal{C}(x)|} \sum\limits_{c \in \mathcal{C}(x)} n(c)$
$c_{min} \leftarrow \underset{\substack{c \in \mathcal{C}(x) \backslash \mathcal{P}(x) \\ n(c) < \overline{n}}}{\mathrm{argmin}} \ n(c)$
if c_{min} is not unique, choose one of the candidates uniformly at random as c_{min}
$m \leftarrow \{y|y \in I(x) \cup \{x\}, c_{min} \in \mathcal{C}(y)\}|$
$c_{max} \leftarrow \underset{c \in \mathcal{P}(x)}{\mathrm{argmax}} \ n(c)$
changeflag $\leftarrow 0$
if $n(c_{max}) > \overline{n}$ and $n(c_{max}) > n(c_{min}) + 1$ **then**
    $p \leftarrow \frac{\overline{n} - n(c_{min})}{m}$
    if $c_{max} = c(x)$ **then**
        $p \leftarrow \frac{p}{2}$
    end if
    $R \leftarrow$ random number uniformly distributed between 0 and 1
    if $R < p$ **then**
        $\mathcal{P}(x) \leftarrow (\mathcal{P}(x) \setminus \{c_{max}\}) \bigcup \{c_{min}\}$
        changeflag $\leftarrow 1$
    end if
end if
if changeflag **then**
    cancel x's running R-channel assignment timer and reschedule to invoke an R-channel selection
end if

the timer expires). However, this behavior can be altered if necessary.

We remark that our algorithm for pool-management bears similarity to the algorithm for minimum conflict coloring in [32]), and the algorithm for channel assignment in Net-X [64]. Also related is the probabilistic distributed learning algorithm for channel assignment described in [72].

Ideally, we would like the pool membership to stabilize after a brief period of churn, with further changes occurring rarely. However, due to the distributed and probabilistic nature of the algorithm, the channel pool membership can exhibit quasi-stable behavior, i.e., after a brief initial period of pool-adjustment, the pool membership may either fully stabilize, or it may largely stabilize with occasional pool membership changes still happening at relatively low rate.

It is to be noted that it is important to introduce some probabilistic damping in the pool-management procedure to achieve good stability properties. One can conceive of many possible formulations for the damping probability, which can aim at reducing the possibility of many interfaces including or evicting the same channel at around the same time. What we use in the current design (see Algorithm 1) is one such formulation, which intuitively tries to reduce the possibility of the same channel being included in the pools of many nearby interfaces at around the same time. Other possibilities include the damping probability formulation used in [64] for channel-assignment, which intuitively tries to reduce the possibility of nearby interfaces on the same channel switching to different channels at around the same time (and can be suitably modified and applied to channel pools). Since the pools are initially chosen uniformly at random, the decisions only involve a two-hop view, and they occur in a staggered manner (due to the initial desynchronization), the protocol performance with many such variant formulations is expected to be similar, since the pool membership would typically adjust and becoming stable or quasi-stable after a brief post-startup period of churn.

R-Channel Selection The R-channel selection algorithm is designed on the premise that all selection decisions are sequential and staggered at different nodes.

To reduce the chance of inadvertant synchronization, the protocol incorporate an element of random jitter in the assignment-interval. Thus, each interface has a R-channel re-assignment timer that is rescheduled over duration $T_{rassign} + X$, where X is a random

variable uniformly distibuted over $[0, J_{rassign}]$.

For simplicity, we currently use globally constant values for $T_{rassign}$ and $J_{rassign}$.

The channel cost metric for channel b computed for interface x of node u has four components:

1. Explicitly known interference conflict cost:

$$C_{einc}(x, b) = \frac{1}{T_{rassign}} \sum_{v \in nbd_2(u)} \sum_{\substack{y \in M_R(v) \\ c(y) = b}} \eta(y) \tag{6.4}$$

2. Interface conflict cost

$$C_{ifc}(x, b) = \frac{1}{T_{rassign}} \sum_{\substack{v \in nbd(u) \\ q_{nbr}(v,u) > 0}} (\chi(v, u, b) - D(v, u, b, x))_+ \tag{6.5}$$

where $D(v, u, b, x) = \frac{q_{nbr}(v,u)}{\max\{\sigma(v,u) - r(v,u,c(x)) + r(v,u,b), \theta\}}$ if $c(x) = b$ or if $[(c(x) \notin c(M_R(v))) \wedge (b \notin c(M_R(v)))]$, and is 0 else.

The intuition behind subtracting $D(v, u, b, x)$ from $\chi(v, u, b)$ is that the latter may sometimes include traffic intended for interface x. This should not be counted as a cost as is, as even after a channel switch, one might typically expect the same amount of traffic (in bits) to be re-directed to whatever new channel x may switch to (although rate difference between the channels should be considered). We also remark that the specific definition of $D(v, u, b, x)$ is driven by the fact that any R-interface x is single-mode, and thus all channels in $\mathcal{C}(x)$ can be operated on by exactly the same set of T-interfaces at a neighbor v.

3. Contention cost (this component helps capture interference beyond the two hop neighborhood which is not captured by the explicit interference cost, and also captures interference conflicts not reflected in queue-lengths):

Let $w_v = q_{nbr}(v, u) + s(v, u)$

$$C_{iinc}(x, b) = \begin{cases} \frac{37.5}{T_{rassign}} \left(\frac{1}{\sum_{v \in nbd(u)} w_v} \sum_{v \in nbd(u)} w_v \kappa(v, b) \right) & \text{if } \sum_{v \in nbd(u)} w_v > 0 \\ 0 & \text{else} \end{cases} \tag{6.6}$$

4. Expected cost of traffic incoming to itself:

$$C_{self}(x, b) = \frac{Incoming(x, b)}{T_{rassign}} \tag{6.7}$$

The cost of a channel b, as computed by R-interface x of node u is given by:

$$Cost(x, b) = C_{einc}(x, b) + C_{ifc}(x, b) + C_{iinc}(x, b) + C_{self}(x, b) \tag{6.8}$$

The R-channel is selected using the procedure in Algorithm 2, which returns the chosen channel. If the chosen channel is different from the current channel, a switch is initiated.

6.5.3 Packet Scheduling: Channel and Interface Binding

The channel and interface selection decisions are decomposed into two separate decisions, viz., channel selection, and interface selection, which are coupled through the channel queue occupancies, and the local interface conflict score χ_{local} (which is a function of the channel queue occupancies, and the number/type of interfaces available at the node).

The channel binding decision is performed by a channel scheduler (denoted by CH-scheduler), and the interface binding decision is performed by an interface scheduler (denoted by IF-scheduler).

The structure of the packet scheduling component is depicted in Fig. 6.4.

The link-layer at each node u maintains the following system of queues:

1. **Neighbor Queues:** Each outgoing unicast packet has a next-hop $v \in nbd(u)$, and is enqueued in the queue corresponding to the appropriate neighbor v. The queue at node u for neighbor v is denoted by $Q_{nbr}(u, v)$, while the length of this queue in bits is denoted by $q_{nbr}(u, v)$, and the length in packets is denoted by $q_{nbr}^p(u, v)$.

2. **Channel Queues:** There is a pair of queues for each channel c such that some interface of u can tune to c. These contain packets that have already been bound to channel c (i.e., these packets will be sent on channel c). The first of these is meant to temporarily hold high-priority packets (LL control packets, ARP packets and routing packets). We shall refer to this as the *high priority holding buffer* for the channel. All other packets are enqueued in the second queue. We shall refer to this as the *channel*

140

Algorithm 2 R-Channel Selection Algorithm (Interface x at Node u)

$\mathcal{S} \leftarrow \mathcal{P}(x)$
if last packet on $c(x)$ received more than M seconds ago **then**
  $\mathcal{S} \leftarrow \mathcal{S} \setminus \{c(x)\}$
end if
for all $b \in \mathcal{S}$ **do**
  if $b \in c(M_R(u) \setminus \{x\})$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{b\}$
  end if
end for
if $\mathcal{S} = \phi$ **then**
  evict first channel in pool; replace with any channel d that is not current channel
  of another R-interface
  return d
  if no such channel found, deactivate interface x
end if
for all $b \in \mathcal{S}$ **do**
  compute $Cost(b)$
end for
$b \leftarrow \underset{c \in \mathcal{S}}{\mathrm{argmin}} \; Cost(c)$
if $c(x) \in \mathcal{S}$ **then**
  if $Incoming(x, c(x)) < \delta_{comb}$ and $\overline{\rho}(x) < \delta_{comb}$ and $\rho(x) < \delta_{comb}$ **then**
    { try to do a combinatorial channel selection instead of a cost-based one}
    $\mathcal{B} \leftarrow \mathcal{P}(x)$
    $I(x) \leftarrow$ the set of all R-interfaces of nodes in $nbd_2(u)$.
    for all $d \in \mathcal{B}$ **do**
      $n(d) \leftarrow |\{y|y \in I(x), c(y) = d\}|$
    end for
    for all $d \in \mathcal{B}$ **do**
      if $(d \notin c(M_R(u)) \wedge (n(d) < n(c(x))))$ **then**
        $p \leftarrow \frac{1}{n(c(x))}$
        $R \leftarrow$ random number uniformly distributed between 0 and 1
        if $R < p$ **then**
          return d
        end if
      end if
    end for
  end if
  if $(C_{einc}(x, b) + C_{ifc}(x, b) + C_{iinc}(x, b)) > 1.0$ **then**
    return $c(x)$
  end if
  if $Incoming(x, c(x)) < \delta_{min}$ or $Cost(c(x)) = 0$ or $Cost(b) >= (Cost(c(x)) - \delta_{inertia})$
  then
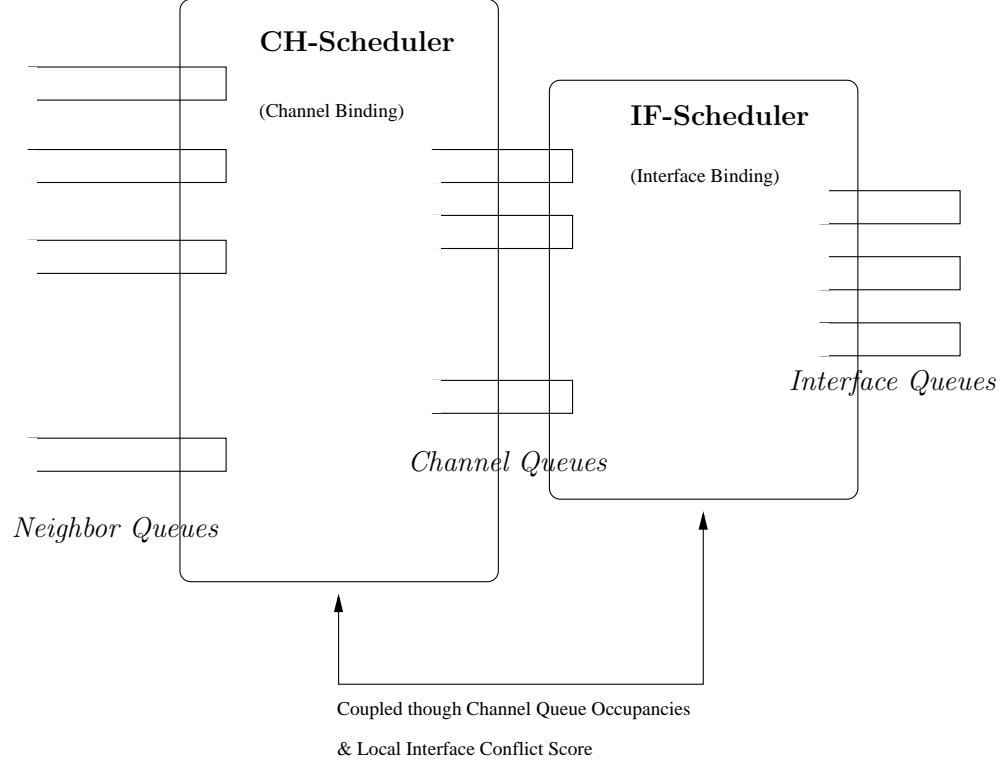   return $c(x)$
  end if
end if
return b

Figure 6.4: Structure of Scheduling Module

queue, and denote this queue for channel c at node u by $Q_{ch}(u, c)$, with the length in bits denoted by $q_{ch}(u, c)$. The length in packets is denoted by $q_{ch}^p(u, c)$.

3. **Interface Queues:** There is a queue for each interface x, containing packets that have already been bound to the interface x, and are awaiting their turn for transmission by interface x. The queue for an interface x is denoted by $Q_{if}(x)$ and the queue-length is denoted by $q_{if}(x)$.

Handling Multi-Channel Broadcast

Currently, we adopt a very simple approach to broadcast. The node v sends a copy of each broadcast packet on all channels that can be operated on by at least one of its interfaces.

High Priority Packets

Broadcast packets have higher priority than unicast packets since typically most of these are expected to be link layer or network layer control packets. Whenever the link layer

receives a broadcast packet for sending, it creates a copy of this packet for each channel and enqueues it in the high-priority holding buffer of that channel.

High-priority unicast packets are handled as follows: if the next-hop node (MAC destination) for the packet is v, the packet is enqueued in the high priority holding buffer of the channel with highest effective rate that can be used to reach that neighbor (i.e., $c(z)$, where $z = \underset{y \in \mathcal{T}(u,v)}{\text{argmax}} r(u, v, c(y)))$.

Link layer control packets also have high priority (note that while LLINFO is broadcast, QINFO and CINFO are unicast). Whenever the link layer generates a control packet to send, it does the following: LLINFO is processed in the same way as other broadcast packets, QINFO/CINFO from u to v are processed like any other high priority unicast packet.

When a routing protocol is in use, it is desirable that any unicast routing control packets should also be given priority.

The CH-scheduler determines how packets will be transferred from the Neighbor Queues to the Channel Queues, while the IF-scheduler determines how packets will be transferred from the Channel Queues to the Interface Queues.

Channel Binding

The CH-scheduler's state at any instant is either blocked or unblocked.

1. Initially, the state is unblocked.

2. Whenever the link layer receives a new packet of regular priority to send from upper layers then, after enqueuing the packet in the appropriate neighbor-queue, it invokes the CH-scheduler.

3. If an invocation of the IF-scheduler results in a non-empty channel-queue becoming empty, the CH-scheduler is invoked after ensuring that its state is unblocked (i.e., if the state is blocked, it is set to unblocked). This is also described later in Section 6.5.3.

4. Whenever the CH-scheduler is invoked:

    (a) If the state is blocked, nothing is done.

    (b) If the state is unblocked, the channel-binding routine (Algorithm 3) is executed. After the execution of the channel-binding routine:

i. If any channel-queue is still empty, the state remains unblocked, else it is set to blocked.

ii. The IF scheduler is invoked.

We now explain the intuition behind the channel binding routine.

A channel-queue is said to be *eligible* for scheduling an invocation of the CH-scheduler if the occupancy of that queue at the time the scheduler was invoked is below a certain threshold $CQ\_THRESH=PKT\_QUANTUM$. Deeming queues with more than CQ_THRESH packets ineligible helps facilitate the objective of late-binding. Queues can also be deemed ineligible if their local conflict score is more than CQ_THRESH.

Consider the set of all eligible neighbor-queues at node u. Each has a certain next-hop node (MAC destination) v for which there is a set of valid interfaces $\mathcal{T}(u,v) \subseteq M_R(v)$, and correspondingly a set of possible channels $\mathcal{T}_c(u,v) = \{c(y)|y \in \mathcal{T}(u,v)\}$.

Since the channel-assignment has already attempted to factor in the traffic-awareness, it is now reasonable to treat the link-layer packet scheduling problem as an independent local decision. From the perspective of the link-layer at node u, each packet enqueued in the set of neighbor-queues has a next-hop node from amongst u's neighbors to which it has to send the packet. Thus, the link-layer treats the local packet scheduling problem as if it were a problem involving single-hop flows.

We draw intuition from the Dynamic Backpressure Scheduler of Tassiulas and Ephremides [110]. In a scenario where all flows traverse only a single-hop, a scheduler which activates links in a manner than maximizes $\sum q_l r_l$ is throughput-optimal (assuming the traffic load falls within the network's stability region). In our scheduling scenario, we can treat each valid (neighbor, channel) pair as a link, and define a conflict between two pairs if they have the same channel. Trying to map the algorithm of [110] directly, one might consider trying to assign packets from various eligible queues to channels, such that the assignment maximizes $\sum q_p \mu_p$, where q_p is the length of the neighbor-queue from which the packet p is taken, and μ_p is the net datarate of the link-channel pair over which p is scheduled.

However, in practice, this can lead to long delays and possible starvation for some flows (especially if some flows are aggressive and inelastic). Additionally, from considerations of amortization, it may be desirable to transfer packets from the neighbor-queues to the channel-queues in certain quanta. An alternative approach might consist of selecting a set

Ω of (neighbor, channel) pairs that maximize $\sum\limits_{v \in \Omega} Age(v)\mu(u,v,c)$, where $Age(v)$ is the age of the HOL (and hence oldest, as the neighbor queues are FIFO) packet of the queue for neighbor v. This gives priority to packets that have been waiting longer, and thus improves fairness characteristics. At the same time, it does not completely deviate from the intuition behind the throughput-optimal dynamic backpressure scheduler described in [110], since a FIFO queue that has been consistently large in the recent past is also likely to have an HOL packet of large age. We adopt a similar approach.

The channel-binding procedure is described in Algorithm 3. Note that comparison between ordered pairs $z_1 = (w_1, r_1)$ and $z_2 = (w_2, r_2)$ is defined as $z_1 > z_2$ if either $w_1 > w_2$ or $w_1 = w_2$ and $r_1 > r_2$; $z_1 = z_2$ if $z_1 \not> z_1$ and $z_2 \not> z_1$.

Interface Binding

The interface binding (IF) scheduler's state at any instant is either blocked or unblocked.

1. Initially, the state is unblocked.

2. Whenever the link layer receives a new broadcast packet, or a high priority unicast packet to send (either a LL control packet, or from upper layers) then, after enqueuing the packet in the appropriate channel-queue (as described in Section 6.5.3), it invokes the IF-scheduler.

3. Whenever an interface-queue becomes empty, a link-layer callback is invoked, which sets the state of the IF-scheduler to unblocked, and invokes it.

4. The IF-scheduler is also invoked after any invocation of the CH-scheduler (as described in Section 6.5.3).

5. Whenever the IF-scheduler is invoked:

    (a) If the state is blocked, nothing is done.

    (b) If state is unblocked, the interface-binding routine (Algorithm 4) is executed. After the execution of the interface-binding algorithm:

        i. If there is no available interface y such that $q_{if}(y) = 0$, the IF-scheduler's state is set to blocked.

Algorithm 3 Channel Binding Algorithm (Node u)

$CQ\_THRESH \leftarrow PKT\_QUANTUM$

for all $v \in nbd(u)$ **do**

  $\mathcal{T}_c(u,v) \leftarrow \bigcup_{y \in \mathcal{T}(u,v)} c(y)$

end for

$\mathcal{S} \leftarrow \bigcup_{v \in nbd(u)} (\{v\} \times \mathcal{T}_c(u,v))$

for all $(v,c) \in \mathcal{S}$ **do**

  if $q_{ch}^p(u,c) > CQ\_THRESH$ or $\chi_{local}(u,c) > CQ\_THRESH$ or $\mu(u,v,c) = 0$ **then**

    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(v,c)\}$

  end if

end for

for all $(v,c) \in \mathcal{S}$ **do**

  if $q_{nbr}(u,v) = 0$ **then**

    $w(v,c) = 0$

  else

    $Age(v) \leftarrow$ time in queue spent by HOL packet of $Q_{nbr}(u,v)$

    $w(v,c) \leftarrow Age(v)\mu(u,v,c)$

    $r'(v,c) \leftarrow \mu(u,v,c)$

  end if

end for

while $\mathcal{S} \neq \phi$ **do**

  $(z,d) \leftarrow \underset{\mathcal{S}}{\operatorname{argmax}} \; (w(v,c), r'(v,c))$

  if $q_{nbr}(u,z) = 0$ **then**

    continue

  end if

  Transfer $\min\{q^p(u,z), PKT\_QUANTUM\}$ packets from $Q_{nbr}(u,z)$ to $Q_{ch}(u,d)$

  for all $(w,b) \in \mathcal{S}$ such that $b = d$ **do**

    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(w,b)\}$

  end for

  for all $(w,b) \in \mathcal{S}$ **do**

    if $\chi_{local}(w,b) > CQ\_THRESH$ **then**

      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(w,b)\}$

    end if

  end for

end while

ii. If some initially non-empty channel queue became empty as a result of packet-transfer during interface binding:

- If the CH-scheduler's state is blocked, it is changed to unblocked.

- The CH-scheduler is invoked.

Note that an interface is deemed to be available for scheduling by the IF-scheduler if it is neither off nor in the process of switching. Also note that the interface-queue lengths may change in the course of execution of the procedure, as packets get transferred.

Interface Queues

Once a packet has been transferred to an interface queue, the link layer relinquishes control over it (except for possibly triggering a flushing of packets from the interface-queue in case of a channel-switch). Whenever an interface-queue becomes empty, a link-layer callback is invoked, which sets the state of the IF-scheduler to unblocked, and invokes it.

6.6   Evaluation

The ns-2 simulator (version 2.31) [46] has been used as the codebase, with substantial modifications to the physical layer and node models. A SINR threshold based model is used, whereby a packet is received successfully if it is received at a power-level equal to or greater than the receiver sensitivity, and the SINR is equal to or greater than the SINR threshold. While this leads to a 0/1 model of packet reception, and does not capture the relationship between SINR and BER, it provides a reasonable approximation for evaluation of a link layer channel and interface management scheme. Cumulative interference has been modeled, and the total received power at an interface used in SINR determination is the sum of the received powers from all packets on the air in that channel at that instant, as well as a small thermal noise component (which is constant for any given channel).

Various rate-specific parameter values used in the evaluation are listed in Table 6.6. The RX-sensitivity values are obtained from the specifications of the Cisco Aironet NIC, while the SINR threshold values are from [126]. A fixed transmission power of 65 mW is used. A data payload size of 1450 bytes is used for all data packets sent. No MAC-layer fragmentation is performed. The carrier-sense threshold is set to -108 dBm (the physical

---
Algorithm 4 Interface Binding Algorithm (Node u)

{First we handle high priority packets}
for all $x \in M_R(u)$ **do**
  if $q_{if}(x) = 0$ and x is available **then**
    transfer packets from high priority holding buffer of $c(x)$ to $Q_{if}(x)$
    till either former is empty, or latter is full
  end if
end for
$\mathcal{C} \leftarrow$ set of all available channels
for all $c \in \mathcal{C} \setminus c(M_R(u))$ **do**
  for all $x \in M_T(u)$ **do**
    if x can operate on c and $q_{if}(x) = 0$ and x is available **then**
      transfer packets from high priority holding buffer of c to $Q_{if}(x)$
      till either former is empty, or latter is full
    end if
  end for
end for
{Next we handle regular priority packets}
for all $x \in M_R(u)$ **do**
  if $q_{if}(x) = 0$ and x is available **then**
    if $q_{ch}(u, c(x)) > 0$ **then**
      Transfer $\min\{q_{ch}^p(u, c(x)), PKT\_QUANTUM\}$ packets from $Q_{ch}(u, c(x))$ to $Q_{if}(x)$
    end if
  end if
end for
$\mathcal{S}(c)$ is the set of T-interfaces of u that can operate on channel c
$\mathcal{S} \leftarrow \{(c, x) | c \in \mathcal{C} \setminus c(M_R(u)), x \in \mathcal{S}(c)\}$
for all $(b, x) \in \mathcal{S}$ **do**
  $w(b, x) \leftarrow$ time in queue spent by HOL packet of $Q_{ch}(u, b)$
  $s'(b, x) \leftarrow -1*$(time to switch from $c(x)$ to b)
  if $(q_{if}(x) > 0)$ or $(\exists\, y \in M_T(u)$ such that $c(y) = b$ and $q_{if}(y) > 0)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(b, x)\}$
  end if
end for
while $\mathcal{S} \neq \phi$ **do**
  $(d, y) \leftarrow \underset{\mathcal{S}}{\mathrm{argmax}}\ (w(b, x), s'(b, x))$
  if $q_{ch}(u, d) = 0$ **then**
    continue
  end if
  Transfer $\min\{q_{ch}^p(u, d), PKT\_QUANTUM\}$ packets from $Q_{ch}(u, d)$ to $Q_{if}(y)$
  for all $(b, x) \in \mathcal{S}$ such that $x = y$ **do**
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(b, x)\}$
  end for
  for all $(b, x) \in \mathcal{S}$ such that $b = d$ **do**
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(b, x)\}$
  end for
end while

Rate	RX Sensitivity	SINR Threshold
1 Mbps	-94 dBm	-2.92 dB
2Mbps	-93 dBm	1.59 dB
6 Mbps	-87 dBm	6.02 dB

Table 6.2: Simulation Parameters

carrier-sense function deems the channel idle if the received power (not considering the thermal noise component) is less than the carrier-sense threshold; thus, the stated carrier-sense threshold should be interpreted as the power that must be received over and above the thermal noise to deem the channel busy). The threshold is deliberately chosen to be much smaller than the receiver sensitivity values, as the resultant carrier-sense range is well beyond 2 hops in our test topologies, and this allows us to evaluate the effectiveness of the protocol in performing channel management with explicit information from 2 hops, when channel conflicts extend beyond this range.[6]

For TCP simulations, the TCP Sack1 agent in ns-2 is used. The initial timeout value has been changed from the ns-2 default, and set to $1.0s$.

The protocol has been evaluated using a set of test topologies, which involve various different kinds of interface configurations and traffic patterns, and facilitate understanding of the strengths and weaknesses of the protocol. Each plotted point on the graphs is an average of 30 independent runs, and the 95% confidence intervals are also plotted.

In all the simulations, we have an initial quiescent period of $40s$ duration to allow the pool-membership to stabilize, before any data transmissions begin. The maximum length of any data session in the simulations is $10s$. We have intentionally chosen a short data session length, as this poses a more difficult case for the protocol, which must be able to adapt to the traffic at a sufficiently fast pace to provide improved performance with short session lengths.

6.6.1   Test Topologies

We use the TwoRayGround propagation model for these topologies, as the primary goal is to study the link layer's ability for dynamic adaptation to traffic in the presence of

---

[6]Note that in this work we are not concerned with choosing a carrier-sense threshold value that is optimal for performance. Our goal is only to evaluate the performance of our protocol given some value for this parameter, and a large carrier-sense threshold poses a more difficult case for our protocol.

Parameter Name	Value
T_{LLINFO}	$0.5s$
J_{LLINFO}	$1.0s$
T_{QINFO}	$0.75s$
J_{QINFO}	$0.25s$
T_{pool}	$4.0s$
$T_{rassign}$	$0.75s$
$J_{rassign}$	$1.0s$
NBR_TTL	$10.0s$
IFR_TTL	$10.0s$
K	1000 (bits)
H	$0.01s$
$\delta_{inertia}$	$\frac{0.1}{T_{rassign}}$
δ_{min}	$\frac{0.1}{T_{rassign}}$
δ_{comb}	0.01

Table 6.3: Protocol Parameter Values Used in Simulations

heterogeneous radios/channels. Results using the probabilistic Shadowing model over some random topologies are discussed in Section 6.6.2. For the choice of simulation parameters used, the TwoRayGround model yields an 802.11a transmission range of approximately 630-640m, and an 802.11g transmission range of approx. 900m. The carrier-sense range is approximately 2130-2140m, which is greater than 3 hops for 802.11a transmissions, and marginally greater than 2 hops for 802.11g transmissions. Note that the ranges obtained with the TwoRayGround model for the chosen parameter settings is larger than what one typically sees in practice; however the absolute value of the transmission range is not very significant for our evaluation.[7]

While discussing the simulation results, we will sometimes refer to the number of channels as c and the poolsize as f. Whenever we show per-flow throughput and the session-durations of different flows are different, the throughput of each flow is computed as total amount of useful data received at the flow destination divided by that flow's session-duration. Whenever we show aggregate throughput, if the session-durations are different for different flows, the aggregate throughout is computed as total amount of useful data

---

[7]However, it is to be noted that the larger propagation delays do have a small effect on the possibility that two nodes within carrier-sense range sense the channel to be idle at around the same time. This sometimes causes a few packet losses due to collisions. However, given the low data rates (and hence less stringent SINR requirements), for certain relative locations of nodes, this can sometimes even improve throughput marginally.
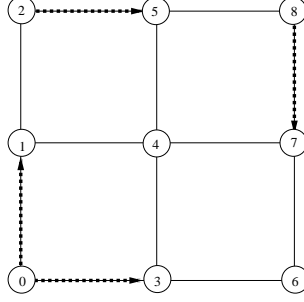
Figure 6.5: Topology 1

received at all flow destinations divided by the maximum session-duration.

Multiple independent runs for each data point were obtained by seeding the defaultRNG object in ns2 with a single selected seed, and then calling the next-substream command i times for the i-th run.

Topology 1   The topology is depicted in Fig. 6.5. 9 nodes are arranged in a 3 by 3 grid (the side of each grid square is 500m). Each node has one R-Interface and one T-interface of type 802.11a. There are 3 CBR flows: $0 \rightarrow 1$ at rate approx. 5.8 Mbps starts at $t = 40.0s$, $0 \rightarrow 3$ at rate approx. 5.8 Mbps starts at $t = 40.5s$, $2 \rightarrow 5$ at rate approx. 2.9 Mbps starts at $t = 40.6s$, $8 \rightarrow 7$ at rate approx. 2.9Mbps starts at $t = 40.9s$. All flows run till end of simulation at $t = 50.0s$. The topology is of interest as it involves both interface and interference conflicts. Note that an ideal scheduler can meet almost all the traffic demand with just 3 channels, by assigning one channel to the R-interface of 0 and either of 1 or 3, assigning the second channel to the remaining node from amongst $1, 3$, and assigning the third channel to 5 and 7. We evaluate the following (number of channels, poolsize) combinations: $(1, 1), (3, 1), (12, 1), (3, 3), (12, 3), (12, 12)$.

The throughput results are depicted in Fig. 6.6. Note that a poolsize of 3 typically yields better performance than a poolsize of 1 for the same number of channels. It is also interesting to note that with 12 channels and poolsize 3, the throughput is lower than the throughput with 3 channels and poolsize 3. The reason for this is that there is an interface-conflict that arises at node 0, as it has only one T-interface but is generating data for both 1 and 3 at ≈ 5.8 Mbps each. Hence it is desirable to have the R-interface of 0 and one of 1 and 3 on the same channel (so that 0 can use its R-interface for transmission), while the T-interface is used to transmit packets to the remaining node on another channel. The interface-conflict
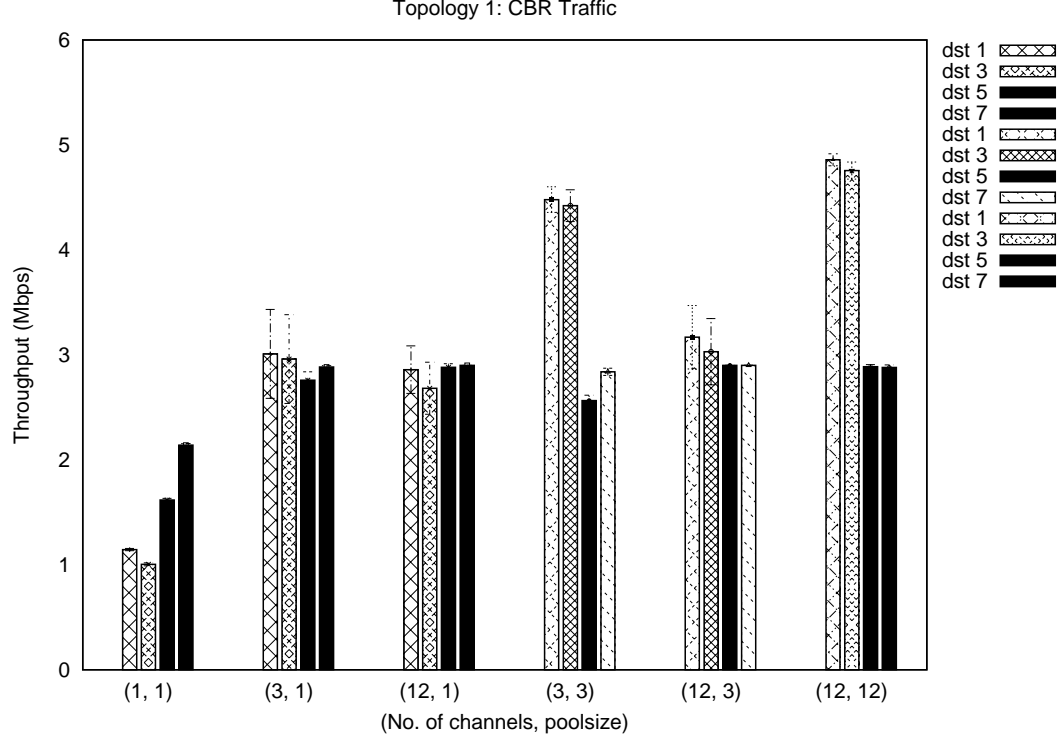

Figure 6.6: Topology 1: CBR Traffic

component of the channel cost metric does try to capture this; however, sometimes the receiver's R-interface cannot change its assignment to address interface conflicts as the transmitter's R-channel may not be in the pool of the receiver's R-channel. This leads to the observed inversion scenario. It can potentially be addressed by additional signaling leading to pool-adjustment, but the extra complexity may not be justified if such scenarios are not very common. Our justification that the inversion phenomenon is being caused by channel-restriction is borne out by the fact that with $(12, 12)$, the throughput is almost the same (actually slightly better) that with $(3, 3)$. The inability to address interface-conflicts is also the cause of the inferior performance with $(3, 1)$ and $(12, 1)$.

The key observation is that $(3, 3)$ and $(12, 12)$ provide close-to-best-possible performance.

Topology 2 8 nodes: 0, 1, ..., 7, are arranged in a linear chain. The separation between adjacent nodes is 500m. Each node is equipped with an 802.11a R-interface and an 802.11a T-interface.

For $K = 1, 2, ..., 7$: We start a single K-hop flow from node 0 to node K at time
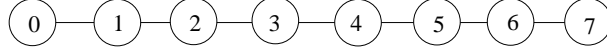
Figure 6.7: Topology 2 (Chain)

$t = 40.0s$, which is active till the end of simulation at $t = 50.s0$.

Fig. 6.8 shows the throughput when the flow comprises CBR (UDP) traffic (generated at approx. 5.8 Mbps). For a given number of channels, setting poolsize (f) to 3 yields better performance that $f = 1$. This is because when $f = 1$, the channel-assignment criterion is solely the number of interfaces on that channel within 2 hops. With a carrier-sense range larger than 2, this may not always achieve good load-balance. Even when the number of channels is large, e.g., $c = 12$, despite the high probability of interfering interfaces having different channels due to sheer randomization, there tend to be a few cases where the channel-assignment is bad, and this degrades the average throughput. This also explains the greater variability (the confidence intervals are larger) with $f = 1$. When $f = 3$, the previously described channel cost metric is used, which includes a contention-cost component that is able to capture high channel load. Thus, even if the channel-assignment is sub-optimal at the time the flow starts, dynamic adaptation to the load occurs, and we get better performance.

Fig. 6.9 shows the throughput when the flow comprises FTP (TCP) traffic.

As can be seen, the throughput with TCP shows a steady decrease as the number of hops increase, even with multiple channels.

While it is true that the LL is better able to adapt to CBR traffic as compared to TCP (since CBR traffic is inelastic, there is a steady queue build-up that eventually triggers channel re-assignment), another major reason for the lower throughput with TCP in the chain topology is the increased delay faced by TCP over multiple hops. As the number of hops to traverse increases, the round-trip delay increases, which has a detrimental effect on TCP throughput.Also note that the performance of almost all the multi-channel combinations is very similar, although one can discern a semblance of relative trends similar to the CBR case. The lack of differentiation can be attributed to the fact that the decline in throughput as the number of hops increase tends to mask the differences due to channel-adaptation.

We remark that the round-trip delay is substantially inflated by the fact that TCP flows have bi-directional traffic (DATA and ACK), and thus at each hop the DATA and
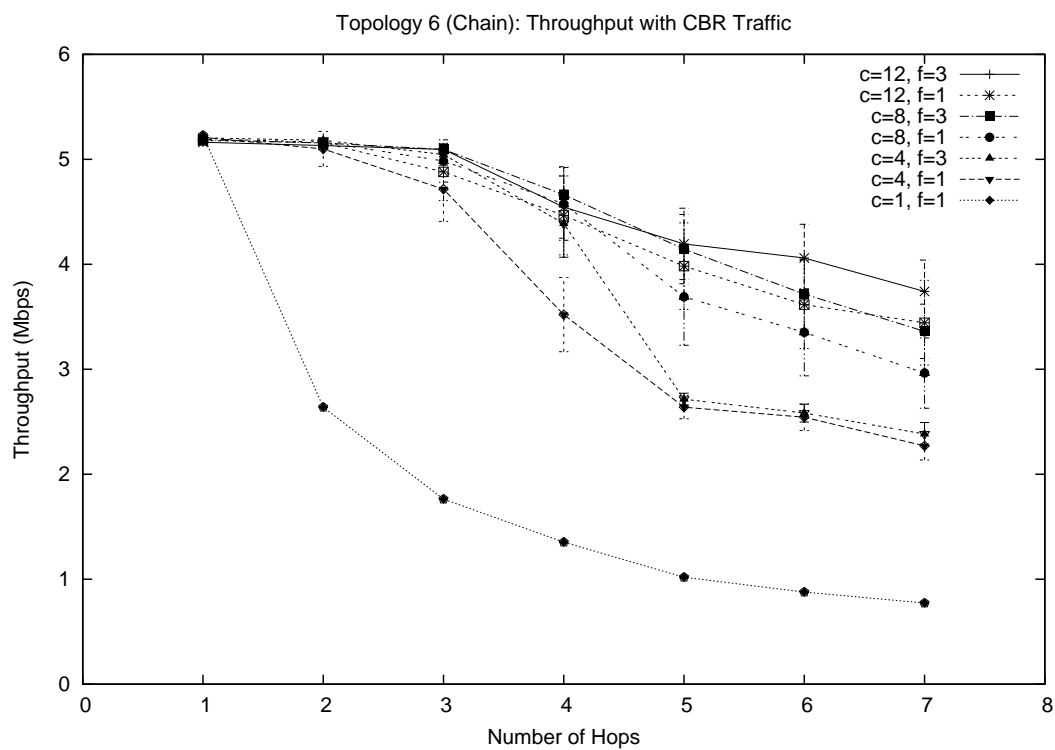
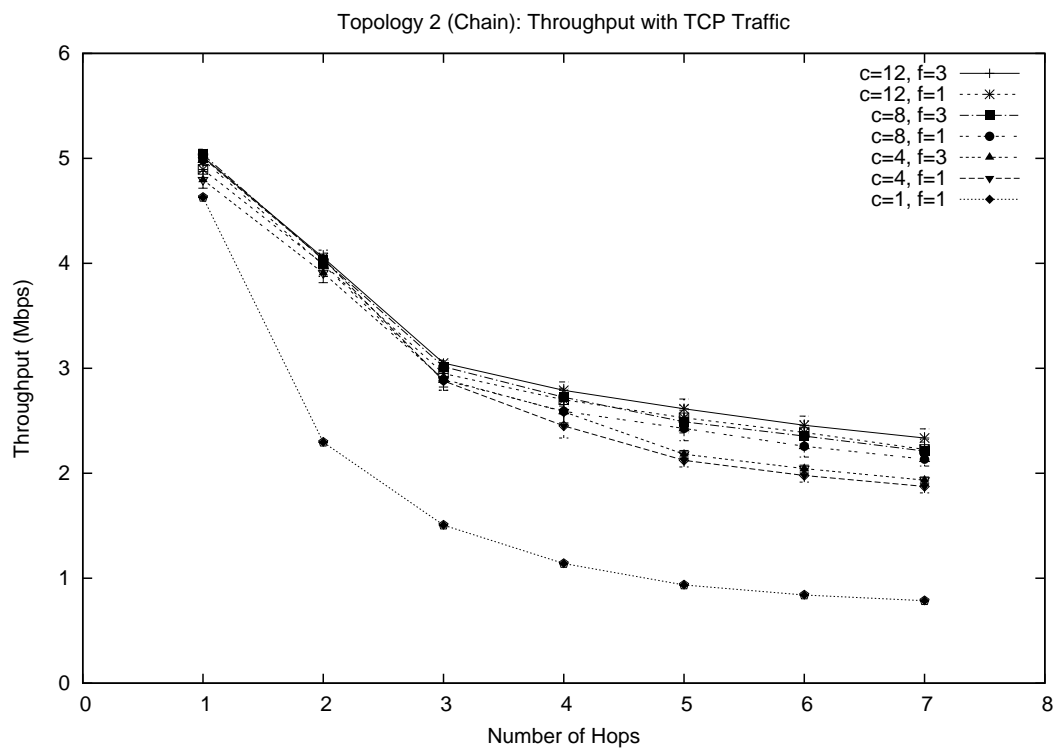Figure 6.8: Topology 2: CBR Traffic
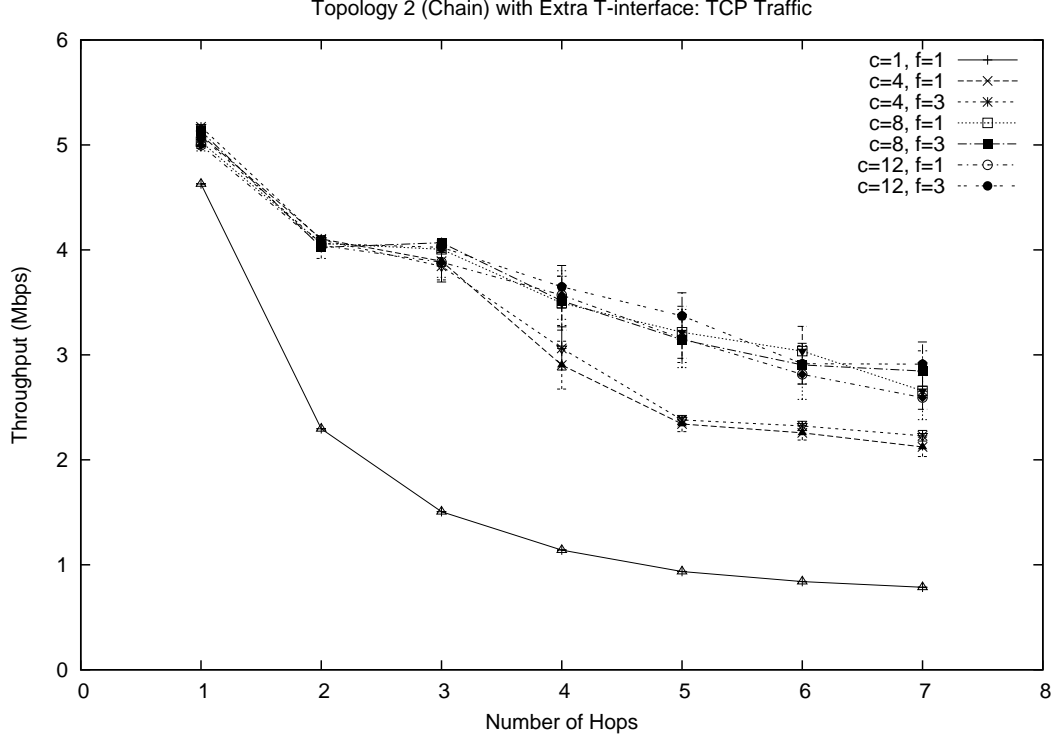

Figure 6.9: Topology 2: TCP Traffic

154

Figure 6.10: Topology 2: (Extra T-Interface): TCP Traffic

ACK packets must share the same T-interface. As evidence of the dominant effect of delay due to DATA/ACK contention, consider a variant scenario where we have the same chain topology, but each node is equipped with an extra 802.11a T-interface. The thoughput results with TCP traffic are shown in Fig. 6.10. It is evident from the figure that the decrease in throughput with increase in hops now occurs at a much slower rate. A similar experimental observation about the improvement in TCP when using additional interfaces for sending was made in the context of the Net-X testbed in [104].

Topology 3 25 nodes are arranged in a 5 by 5 grid spatial layout (the side of each grid square is 460m). Thus, the logical network topology is also a 5 by 5 grid. Each node is equipped with one pair of 802.11a interfaces (one R-interface and one T-interface). We pre-designate 12 (disjoint) one-hop SD pairs, as depicted in Fig. 6.11. We vary the number of channels c. If c channels are in use, the first c sources start sending data at $t = 40.0s$ and continue till the end of simulation at $t = 50.0s$. Thus, the number of flows in any scenario is the same as the number of channels. Therefore, an ideal omniscient scheduler can assign each flow to a separate channel, and get the full benefit of each channel, providing maximum
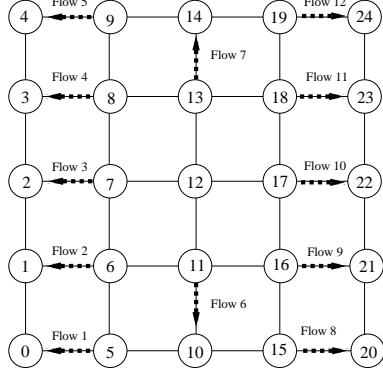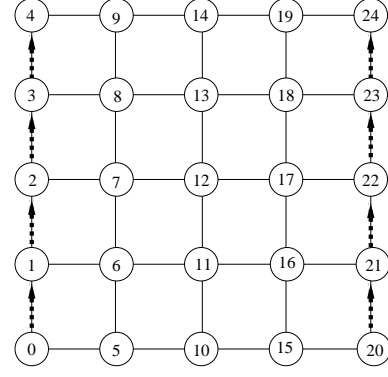
Figure 6.11: Topology 3
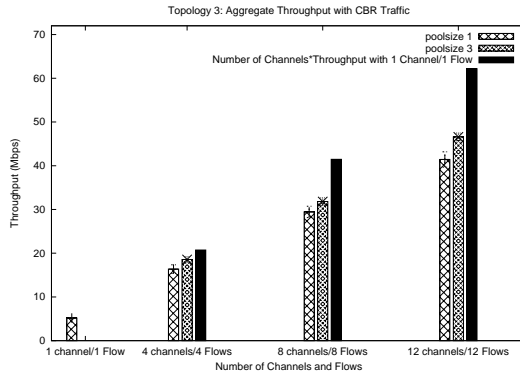

Figure 6.12: Topology 4
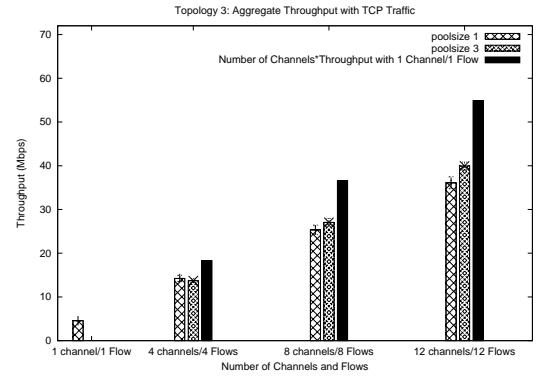

Figure 6.13: Topology 3: CBR Traffic

Figure 6.14: Topology 3: TCP Traffic

throughout to each flow. However, we have a distributed protocol where each node only has explicit information up to a two-hop neighborhood, and has reduced flexibility due to channel-restriction. Thus, this topology provides a means of evaluating the efficacy of the protocol in adapting the channel of an interface to traffic that may extend beyond its two-hop neighborhood.

At time $t = 40.0s$, all c active sources start sending to their respective destinations, and continue to do so till the simulation ends at $t = 50.0s$.

Fig. 6.13 depicts aggregate throughput for CBR traffic. Given c channels, a useful benchmark is to compare the achieved throughput with c times the single-channel throughput. While the difference between this and what the LL is able to achieve increases as c increases, one can see that even with $c = 12$, the LL is able to get quite good performance. Also $f = 3$ shows a small but consistent performance gain over $f = 1$.

Fig. 6.14 depicts aggregate throughput for TCP traffic. The relative trends are similar, although the throughput obtained is lower than in the case of CBR traffic.
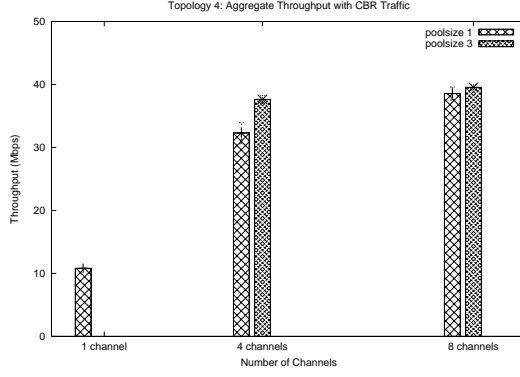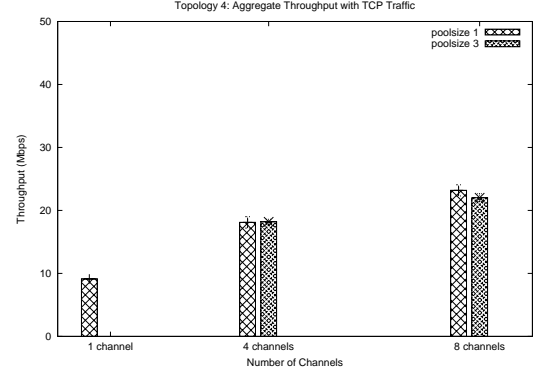
Figure 6.15: Topology 4: CBR Traffic
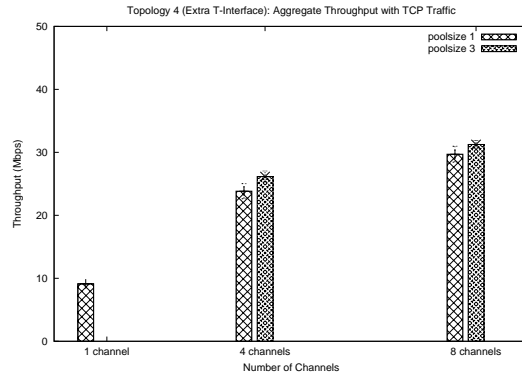

Figure 6.16: Topology 4: TCP Traffic

Figure 6.17: Topology 4 with Extra T-interface: TCP Traffic

Topology 4 25 nodes are arranged in a 5 by 5 grid layout (the side of each grid square is 600m). Thus, the logical network topology is also a 5 by 5 grid. Each node is equipped with one pair of 802.11a interfaces (one R-interface and one T-interface). We pre-designate 8 (disjoint) one-hop SD pairs, as depicted in Fig. 6.12, in the two extreme columns of the grid. All sources start transmitting at $t = 40.0s$ and continue till the end of simulation at $t = 50.0s$. Given the grid-size, it can be seen the all sources within the same grid column are within each others' carrier-sense range, but the sources in different columns are not. This yields a spatial reuse factor of 2 for up to 4 channels. Thus, an ideal scheduler needs just 4 channels to be able to concurrently schedule the flows. We evaluate the efficacy of the protocol in handling this situation.

Fig. 6.15 depicts the aggregate throughput when all flows comprise CBR traffic at rate approx 5.8 Mbps each. As can be seen, even with just 4 channels, the performance with poolsize 3 is very close to what we would expect from an ideal scheduler. A poolsize of 1 with 4 channels yields a performance that is moderately but not drastically inferior to using
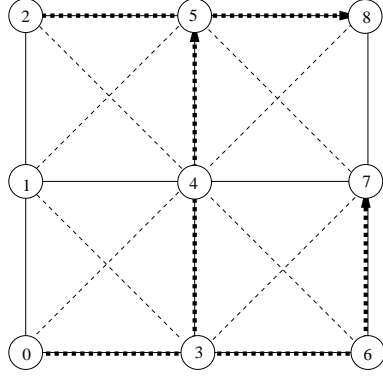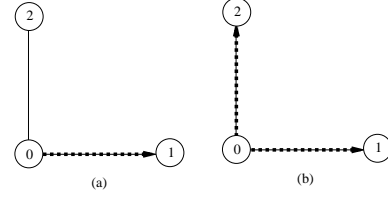
Figure 6.18: Topology 5

Figure 6.19: Topology 6

a poolsize of 3. With 8 channels, the performance is almost the same for poolsize 3 (as the performance of poolsize 3 with 4 channels is already close to the best possible, there is little margin for improvement). However, with 8 channels, poolsize 1 also performs almost as well, since the number of channels is sufficiently larger than the number of mutually conflicting flows.

Fig. 6.16 depicts the aggregate throughput when all flows comprise FTP traffic. In this case, we see that with 4 channels, the throughput is little better than twice the throughput with 1 channel. Increasing the number of channels to 8 yields only marginal gain. Once again, we remark that the LL is less effective in dynamically adapting the channel assignment to TCP traffic, and this can explain the lower throughput with TCP to some extent. However, the rather poor performance with TCP is also due to the fact that the flow-endpoints are not disjoint. As can be seen, the destination of flow 1 is the source of flow 2, and so on. Resultantly, these nodes have to share their T-interface between DATA for one flow, and ACK for another. Thus, the phenomenon is similar to what we discussed in the context of the chain topology. To verify this, we equipped each node with an extra T-interface, and performed the simulation for FTP traffic. The aggregate throughput is depicted in Fig. 6.17, and shows substantial improvement over the previous case.

Topology 5   This topology (Fig. 6.18) helps evaluate how the link layer schedules packets over different channels and interfaces, given multi-hop flows with routes specified as sequences of nodes. 9 nodes are arranged in a 3 by 3 grid layout (the side of each grid square is 500m). Thus, the 802.11a induced topology is a 3 by 3 grid, but the 802.11g links span diagonals. Each node has one R-Interface and one T-interface of each type 802.11a
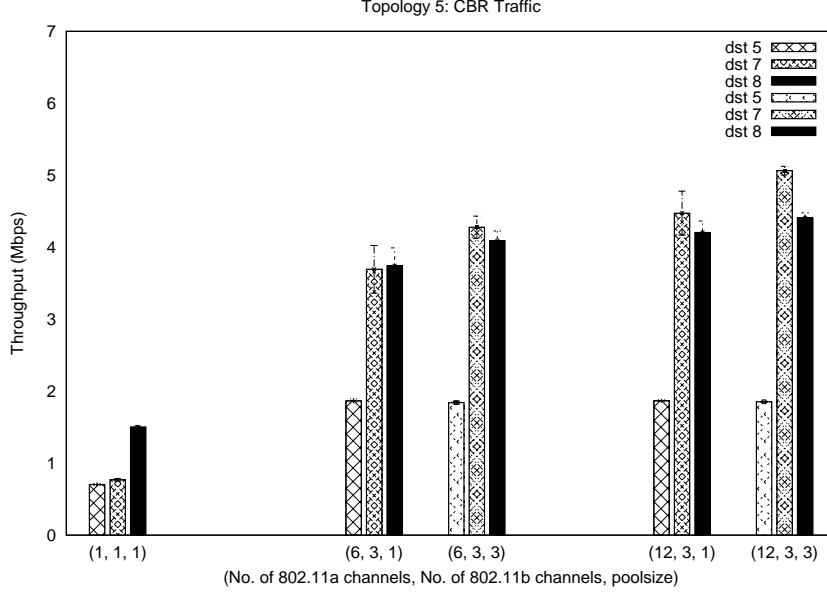
Figure 6.20: Topology 5: CBR Traffic

and 802.11g.

There are 3 flows: $0 \rightarrow 7$ with manually specified route $0 \rightarrow 3 \rightarrow 6 \rightarrow 7$, $3 \rightarrow 5$ with manually specified route $3 \rightarrow 4 \rightarrow 5$, and $2 \rightarrow 8$ with manually specified route $2 \rightarrow 5 \rightarrow 8$.

In the CBR traffic case, the traffic generation rates are : $0 \rightarrow 7$ at rate approx. 5.8 Mbps , $3 \rightarrow 5$ at rate approx. 2 Mbps, and $2 \rightarrow 8$ at rate approx. 5.8 Mbps. Note than an ideal scheduler can meet almost all the traffic demand with just 5 802.11a channels, and 2 802.11g channels.

We evaluate performance with the following combinations of (number of 802.11a channels, number of 802.11g channels, poolsize): $(1, 1, 1), (6, 3, 1), (6, 3, 3), (12, 3, 1), (12, 3, 3)$.

Fig. 6.20 depicts the per-flow throughput with CBR flows. It can be seen that $(6, 3, 3)$ and $(12, 3, 3)$ perform very well, and yield throughput fairly close to what we would expect in the best case. This indicates that the LL is able to adjust the channel assignment as per the traffic, and is also able to distribute packets across the different types of interfaces/channels in a reasonable manner. For the same number of channels, the performance with $f = 1$ is inferior to that with $f = 3$, due to lack of dynamic R-channel adaptation.

Fig. 6.21 depicts the throughput when all the 3 flows comprise FTP traffic. It can be seen that the throughput is lower than the CBR case, which is to be expected as we have multi-hop TCP flows. All multi-channel combinations have similar performance, as
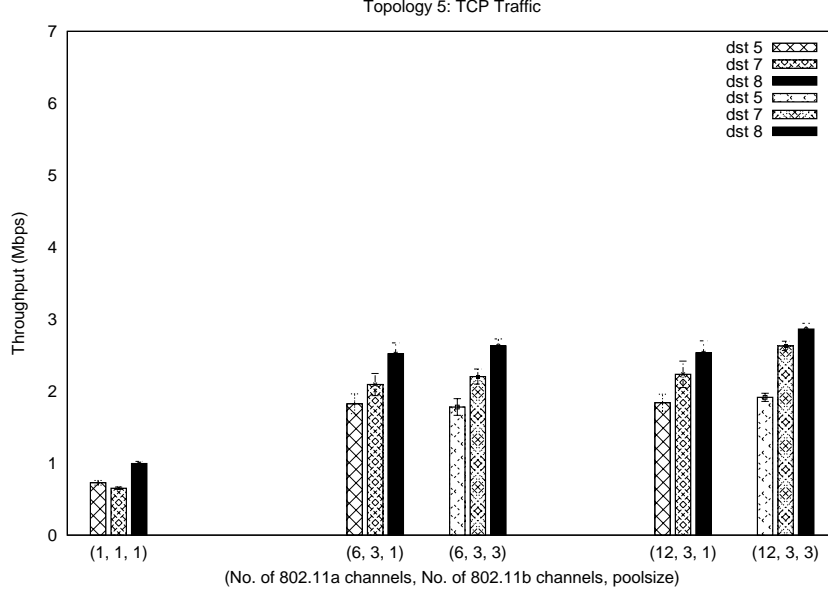
Figure 6.21: Topology 5: TCP Traffic

any differences are likely masked by the degradation in TCP throughput due to traversing multiple hops.

Topology 6 This simple topology (Fig. 6.19) illustrates in detail how the link layer handles packet scheduling, when there are neighbors with different interface types, including multimode T-interfaces. The topology comprises a single-hop network of 3 nodes 0, 1, 2.

We consider the following variant scenarios:

1. *Topology 6.1:* 0 and 1 have one R-interface and one T-interface each of type 802.11a and 802.11g. 2 has one 802.11g R-interface and 1 802.11g T-interface. One flow: $0 \rightarrow 1$. Two traffic scenarios are considered: (i) approx. 7.73 Mbps CBR (ii) FTP

2. *Topology 6.2:* 0 and 1 have one R-interface and one T-interface each of type 802.11a and 802.11g. 2 has one 802.11g R-interface and 1 802.11g T-interface. Two flows: (i) $0 \rightarrow 1$ at approx. 5.8 Mbps CBR , $0 \rightarrow 2$ at approx. 1.93 Mbps CBR (ii) $0 \rightarrow 1$ FTP and $0 \rightarrow 2$ FTP

3. *Topology 6.3:* 0 and 1 have one 802.11a R-interface, one 802.11g R-interface, and 1 802.11ag T-interface. 2 has one 802.11g R-interface and 1 802.11g T-interface. One flow: $0 \rightarrow 1$. Two traffic scenarios are considered: (i) approx. 7.73 Mbps CBR (ii) FTP
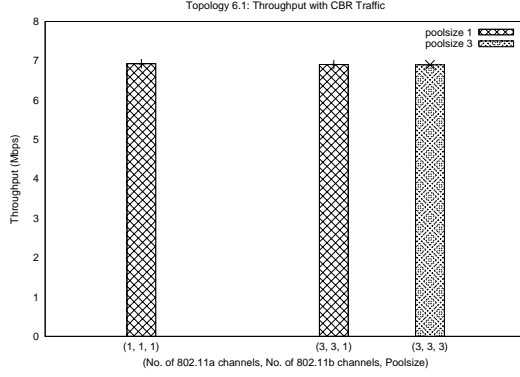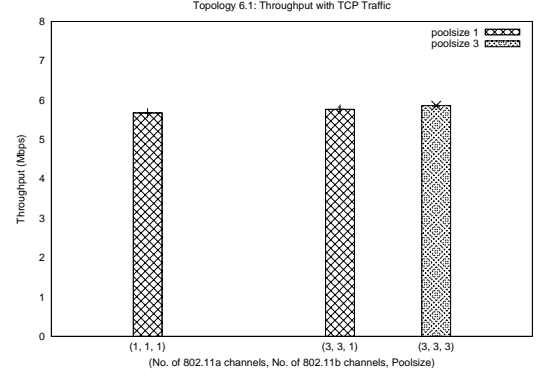
160

Figure 6.22: Topology 6.1: CBR Traffic

Figure 6.23: Topology 6.1: TCP Traffic

4. *Topology 6.4:* 0 and 1 have one 802.11a R-interface, one 802.11g R-interface, and 1 802.11ag T-interface. 2 has one 802.11g R-interface and 1 802.11g T-interface. Two flows: (i) $0 \rightarrow 1$ at approx. 5.8 Mbps CBR , $0 \rightarrow 2$ at approx. 1.93 Mbps CBR. (ii) $0 \rightarrow 1$ FTP and $0 \rightarrow 2$ FTP

In all the above scenarios, the $0 \rightarrow 1$ flow starts at $t = 40.0s$, the $0 \rightarrow 2$ flow starts at $t = 42.0s$ (whenever applicable), and continue(s) till the end of simulation at $t = 50.0s$. We evaluate performance with the following combinations of (number of 802.11a channels, number of 802.11g channels, poolsize): $(1, 1, 1), (3, 3, 1), (3, 3, 3)$.

In all the above topologies, the performance with $(1, 1, 1)$ is very good. When there is only one channel of each type, the LL of each node deactivates the T-interfaces. Resultantly, nodes 0 and 1 effectively have 1 802.11a interface on the single 802.11a channel and 1 802.11g interface on the single 802.11g channel, while node 2 has one 802.11g interface on the one 802.11g channel. Thus, node 0 can simultaneously transmit on both channels to its destination(s).

In Topologies 6.1 and 6.2, node 0 has one T-interface of each type, and thus both the multi-channel combinations also have performance similar to $(1, 1, 1)$.

When node 0 has a single multi-mode T-interface, and there is a single flow (Topology 6.3), $(3, 3, 1)$ exhibits lower performance than the other two combinations. The difference is more marked with CBR traffic (Fig. 6.26), as compared to TCP traffic (Fig. 6.27). This is because with $(3, 3, 1)$, the R-interfaces are more likely to be on different channels, and thus node 0 can only use its multi-mode T-interface to send data. Note that the local interface conflict score helps ensure that the data is primarily sent using the 802.11a channel

161
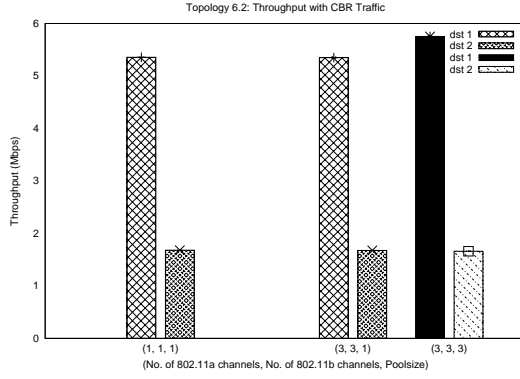
Figure 6.24: Topology 6.2: CBR Traffic
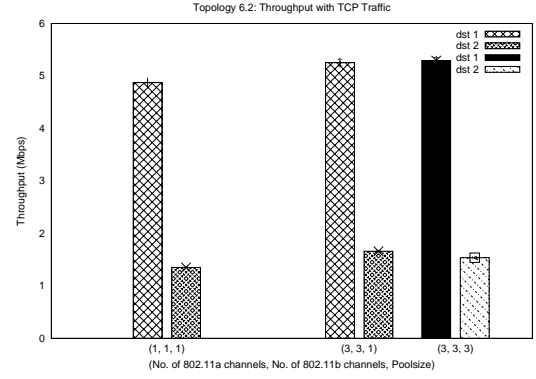
Figure 6.25: Topology 6.2: TCP Traffic
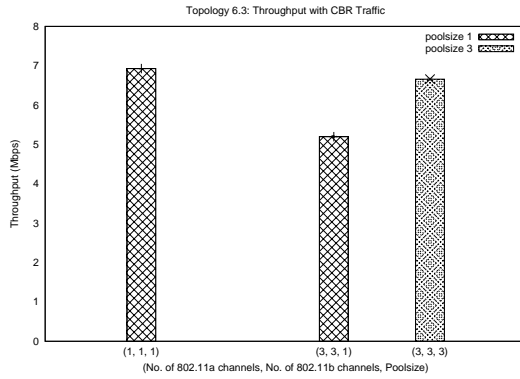
Figure 6.26: Topology 6.3: CBR Traffic
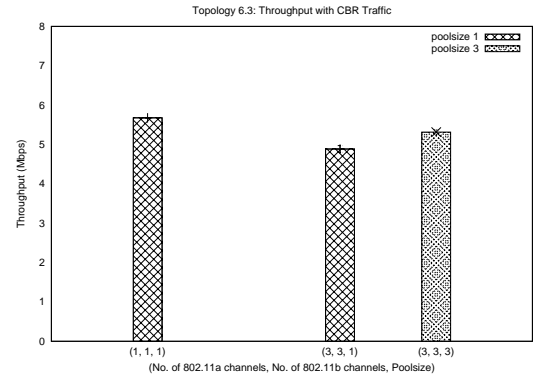
Figure 6.27: Topology 6.3: TCP Traffic

on the multi-mode interface. In case of $(1, 1, 1)$, by default the R-interfaces get used, as explained earlier, and we get the benefit of data-striping across 2 channels. With $(3, 3, 3)$, the network is likely to initially have the R-interfaces on different channels, but is able to quickly adapt based on the interface conflict cost, and get the benefit of data-striping across two interfaces/channels. TCP throughput is typically moderately lower than CBR traffic, and the difference between $(3, 3, 1)$ and $(3, 3, 3)$ is not very marked. This can be explained by the fact that TCP probably gets lesser benefit from data-striping due to out-of-order delivery issues.

When node 0 has a single multi-mode T-interface, and there are two flows (Topology 6.4), $(3, 3, 1)$ again exhibits much lower performance (this time for both CBR and TCP), as there is a smaller chance of R-channel overlap, and thus, node 0 must typically time-share its T-interface to send to node 1 and node 2. The other multi-channel combinations benefit from the R-interfaces, as already explained above. Also note that despite having to contend for the same interface, the two flows each get reasonable throughput. Of course, the
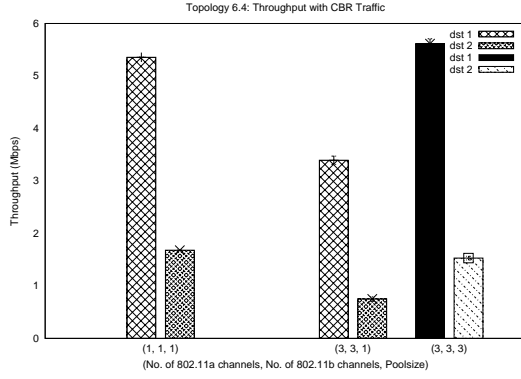
Figure 6.28: Topology 6.4: CBR Traffic
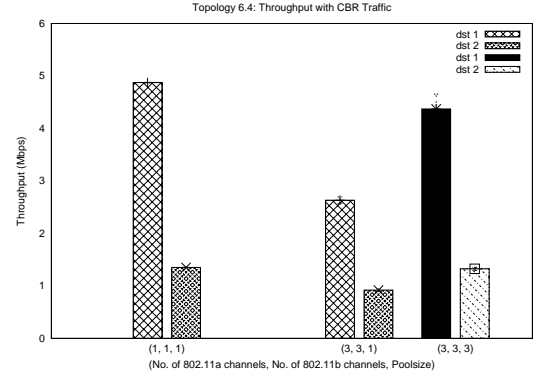
Figure 6.29: Topology 6.4: TCP Traffic

throughput for destination 2 is lower, since the packet-scheduler tries to achieve a balance between providing some fairness and getting the best rate (though the two flows do get a reasonably fair share of interface time). If greater throughput fairness is needed, the scheduling rules can be suitably modified to achieve that.

Topology 7   9 nodes are arranged in a 3 by 3 grid (the side of each grid square is 500m). Node 4 has 4 R-interfaces of type 802.11a, and one T-interface of type 802.11a. All other nodes have one R-interface and one T-interface of type 802.11a. There are 4 one-hop flows $1 \rightarrow 4, 3 \rightarrow 4, 5 \rightarrow 4, 7 \rightarrow 4$ which start at times $t = 40.0s, 40.5s, 41.0s, 41.5s$ respectively, and continue till end of simulation at $t = 50.0s$. In the CBR traffic case, each flow has traffic rate approx. 5.8 Mbps.

The following combinations of (number of channels, poolsize) were simulated: (1, 1), (4, 1), (4, 3), (12, 1), (12, 3), (12, 12).

This topologies are of interest as it involves nodes with a different number of radio-interfaces. Moreover, it is representative of scenarios where node 4 may be a gateway or server node which is likely to be more capable than others, and to which much of the traffic might be directed. It is also of interest as an illustration of how various LL mechanisms complement and supplement each other.

The first observation we make is that an ideal scheduler needs just 4 channels to get best-possible performance (as the receiver has 4 R-interfaces), and it can do so by simply partitioning the 4 senders across channels. However, when each sender independently decides which channel(s) to use, there is the possibility that two or more senders may try to access the same channel at the same time. This would create contention on this channel,

163
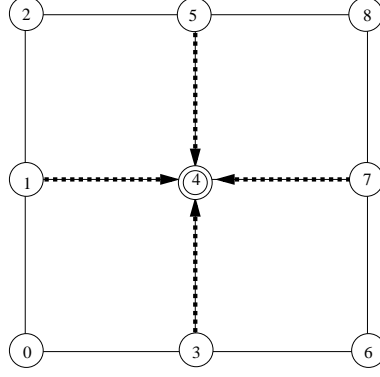
Figure 6.30: Topology 7

while some other channel might be unutilized.

Fig. 6.31 shows the aggregate throughput when all the flows are CBR. Note that all multi-channel combinations give throughput that is fairly close to the best-possible throughput; $(4, 1), (4, 3)$ and $(12, 12)$ provide best performance, but even $(12, 3)$ and $(12, 1)$ are only marginally inferior. However, each combination has some distinct characteristics, as we now explain.

Suppose we did not have an interface conflict cost or an local interface conflict score.

Note that when we have just 4 channels, we would always get very good throughput (for both $(4, 1)$ and $(4, 3)$). The reason is as follows: each of the 4 R-interfaces at node 4 will be on one each of these channels. The R-interfaces at the senders must also be on some of these channels (as there are no other channels). Thus, there is bound to be overlap in the R-channels of senders and receivers. This would allow some/all of the senders to use both their interfaces for sending (since the LL performs data-striping). Moreover, there is likely to be at least one active sending interface on each channel, and we can get good channel utilization and throughput.

Suppose we have an interface conflict cost, but do not have a local interface conflict score.

With $(12, 12)$, all the R-interfaces are initially likely to be on different channels, but after the data sessions start, if the senders are not able to send data fast enough, the queues will build up, and the interface-conflict cost will tend to lead the R-interfaces of node 4 to switch to the R-channels of each of the senders. With $(12, 3)$ such an adaptation is less likely to happen (due to the channel restriction), and throughput would be lower.
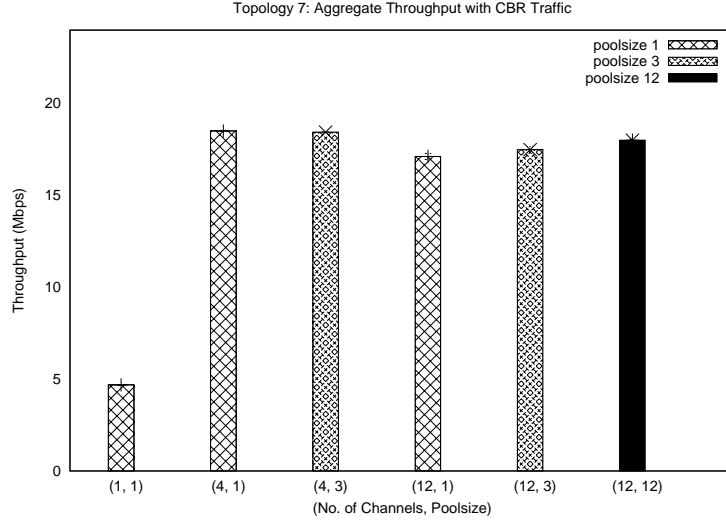
Figure 6.31: Topology 7: CBR Traffic

With $(12, 1)$, there is a very small chance of substantial R-channel overlap a priori, and there is no traffic-dependent R-channel re-assignment to aid this. Moreover, in the absence of a local interface conflict score, the channel-binding algorithm at each sender will tend to bind packets to many different channels (from amongst the 4 choices), if the neighbor-queue is sufficiently large when the CH-scheduler is invoked. This would lead to reduced throughput.

The use of the local interface conflict score helps address this.

Note that each sender has 4 channel choices for sending data, but only one T-interface (assuming there is no overlap in R-channels). Thus, all these 4 channels have a local interface conflict with each other. When the channel binding procedure is executed, packets will be preferentially bound to the channel with highest net datarate, and hence lowest recent contention. Once some packets have been bound to this channel, the local interface conflict would make the other channels ineligible. Therefore, if each sender were to have a different *best* channel, this would lead to a near-partition of senders across channels. Note that by the very nature of the net datarate statistic, a node that recently won quick access to a channel will consider it a good channel, while other senders are likely to find it less attractive. This is likely to lead to the desired scenario. This explains the good performance even with $(12, 1)$ and $(12, 3)$.

Fig. 6.32 shows the aggregate throughput with TCP flows. The trends are similar to the CBR case, although the achieved throughput is generally lower.

Topology 7: Aggregate Throughput with TCP Traffic
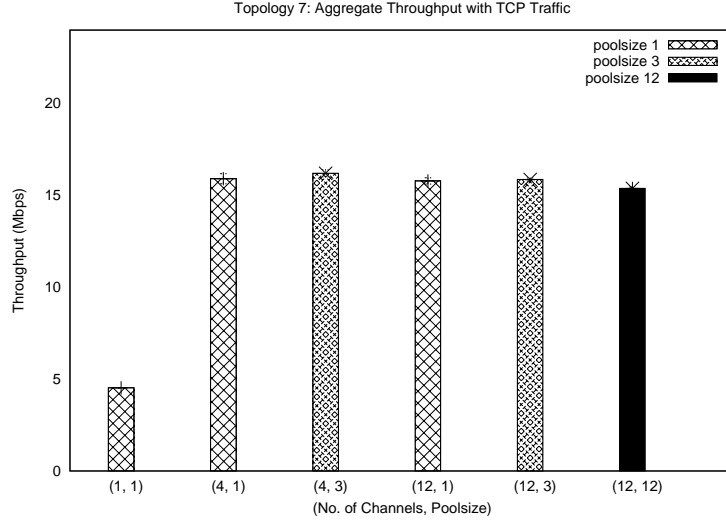
Figure 6.32: Topology 7: TCP Traffic

6.6.2 Random Topologies

To get some insight into performance in a lossy environment, we have performed some simulations on random topologies with the shadowing model.

We considered 10 static random topologies of 30 nodes over a 600mx600m area. The shadowing model in the ns-2 simulator was used with a path-loss exponent of 2.5 and a shadowing deviation of 2dB.

Each node is equipped with an 802.11a R-interface, an 802.11a T-interface, an 802.11g R-interface and an 802.11g T-interface. We pre-designate 12 nodes as potential sources: $s_1 = 0, s_2 = 2, ..., s_{12} = 22$. We consider 2 channel/traffic configurations:

- 1 802.11a channel, 1 802.11g channel, poolsize 1, referred to as $(1, 1, 1)$. At $t = 40.0s$, s_1 chooses a random next-hop node as destination and starts transmitting. It continues to do so till the simulation ends at $t = 50.0s$.

- 12 802.11a channels, 3 802.11a channels, poolsizes 1 and 3, referred to as $(12, 3, 1)$ and $(12, 3, 3)$ respectively. At $t = 40.0s$, all 12 sources $s_1, ..., s_{12}$ choose a random next-hop node as destination and start transmitting. They continue to do so till the simulation ends at $t = 50.0s$.

Thus, in each configuration, the number of flows is the same as the number of 802.11a channels in use.
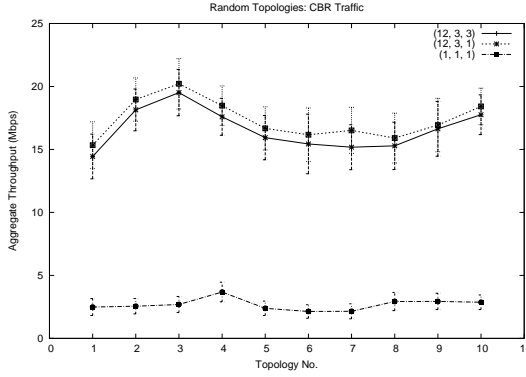
Figure 6.33: Random Topologies: CBR Traffic
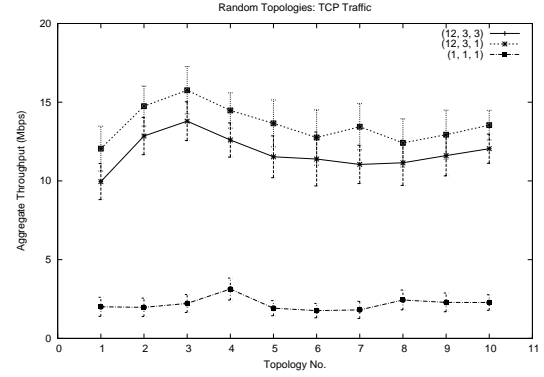
Figure 6.34: Random Topologies: TCP Traffic

To get a random sampling of links from the designated source(s), the random choice of neighboring destination is made at runtime for each run, by inspecting the link layer neighbor-list of the source node, and making a random selection from amongst all symmetric neighbors (without regard to 802.11a reachability). Thus, the link may only be operational on 802.11g. Also, the destinations are likely to be different for each of the 30 runs for each plotted point. Furthermore, as the choice is made dynamically at runtime, there is a small possibility that it may not always be the same for the same run number of different configurations, even though the seed is the same (this can happen if the neighbor-list membership is different at the time of selection, which is not very likely except in very lossy scenarios, i.e., very large shadowing deviation values).

Multiple independent runs for each data point were obtained as follows: for independent run i, the defaultRNG object in ns2 was seeded with a single selected seed (the same for all runs), and then the next-substream command was invoked i times. Each channel has a separate associated Shadowing propagation object in our simulation code; each of these objects also has an associated RNG. These are not explicitly seeded (we have changed the default ns2 behavior), as the ns2 random number generator automatically assigns a seed to each new RNG coresponding to an independent stream, once the defaultRNG has been seeded. However, the next-substream command is invoked i times on each Shadowing RNG for run i. In addition, each node's LL has an RNG which is used for the random destination choice. These are also assigned automatic independent seeds by ns2. The next-substream command is invoked i times on each of these RNGs for run i.

Fig. 6.33 shows the aggregate throughput when all flows are CBR with rate approx. 5.8

167

Mbps. The throughput for $(1, 1, 1)$ is much lower than what we would ideally expect, due to the losses induced by the shadowing model. Similarly, the throughput for $(12, 3, 1)$ and $(12, 3, 3)$ is also much lower than the ideal. However, we do consistently get approximately a 6-7 times improvement over the single-channel case by using multiple channels. While this is certainly far from ideal, given that there are 12 802.11a and 3 802.11g channels, it is actually quite satisfactory, given the nature of the topology and the traffic pattern. Recall that we choose a random neighbor from the neighbor-list of the designated source(s). Thus, often the neighbor may be reachable only using 802.11g (which has higher range). Since, there are only 3 802.11g channels, this can limit the possible improvement. Another observation is that the average aggregate throughput with $(12, 3, 1)$ is marginally but fairly consistently higher than $(12, 3, 3)$, but in most cases, the confidence-intervals overlap substantially, and so the difference is not very relevent statistically. The slightly better performance of $(12, 3, 1)$ can be explained by the fact that $(12, 3, 3)$ does not get much opportunity to gain from dynamic adaptation (if many active links are 802.11g only, then there is not much scope for adaptation; moreover the channel estimation procedure is not very sophisticated, and may thus occasionally initiate unwarranted R-channel changes on perceiving a low effective rate on the current channel), but it does incur some additional overhead since more control data is sent when the poolsize is greater than 1.

Fig. 6.34 shows the aggregate throughput when all flows are FTP, i.e., TCP traffic. The relative trends are similar, though the throughput is lower, and the comparative improvement on using multiple channels is also smaller. This is due to the greater impact of losses on TCP, even leading to flow-starvation sometimes. The difference between $(12, 3, 1)$ and $(12, 3, 3)$ is much more marked, though the confidence intervals still exhibit overlap.

6.7 Discussion

The proposed HMCLL protocol is able to address a wide range of scenarios in a satisfactory manner. It is to be noted that much of the benefit of using dynamic channel adaptation seems to arise in scenarios where there are interface conflicts, or in scenarios with interference conflicts with the number of active links comparable to the number of channels. When there are only interference conflicts and the number of flows is much smaller than the number of channels, even having poolsize 1 (which corresponds to a quasi-static combi-

natorially load-balanced assignment) usually works fairly well. However, having a poolsize greater than 1 does generally help improve consistency even in such situations, and helps avoid the occasional worst-case scenarios that can arise with poolsize 1. The two-level scheduling component provides fairly satisfactory performance. In particular, the coupling introduced by the local interface conflict score helps the LL effectively address scenarios with multi-mode T-interfaces and scenarios where the receiver has many R-interfaces, but the sender may have only one or few T-interfaces.

The LL is able to adapt the channel assignment to CBR traffic much more easily than to TCP traffic. The primary reason for this is that if the network is crrently in a sub-optimal channel assignment configuration, the queues will build up in the CBR case, and when the information propagates within 2 hops, it will likely trigger a R-channel switch at some interface(s) to a less loaded channel. However, with TCP traffic (especially flows that just traverse one-hop), the queues may never become very large, as the source may adjust its rate quickly to the available bandwidth. Thus, the queue may not always build up to the extent needed to trigger a switch (recall that we have an element of hysteresis). To alleviate this, we have included an *excess utilisation* component in the interface conflict cost. We also have an implicit interference-cost element which is based on experienced contention-time, which helps address both the issue of load due to TCP flows, and also load due to any type of traffic which lies byond two hops (as the interface will not have explicit information of this). However, there is still potential for further improvement.

The results for the random topologies with the shadowing model indicate that poolsize 1 is actually marginally better. As mentioned earlier, this can be explained by the fact that there is limited potential for improvement through dynamic adaptation, and having a poolsize greater than 1 implies slightly more overhead, and can also cause some unwarranted channel switches (since the channel estimation procedure is quite rudimentary, and involves little active probing). Thus, there is much potential for improvement along these lines.

6.8 Future Directions

In the course of our work on the described protocol, we have identified certain interesting directions for future work, involving both theoretical and protocol design aspects.

Neighborhood Management   Currently, our protocol makes the implicit assumption that reachability characteristics are the same for all channels in the same band. Thus, if a neighbor is deemed reachable using an 802.11a channel, then the effective-rate for all 802.11a channels on that link is set to be the raw datarate, till some rate-history has been accumulated (as a result of packet transmissions). However, reachability characteristics can be different even for channels in the same band. One reason for this is the possibility of varying levels of external noise. Another reason is that the difference in frequency can lead to different propagation characteristics. While one would expect that within a single band, this difference would not have a significant effect, however, if two nodes are at the fringes of each others' transmission range, a change of R-channel by one neighbor can potentially even make them unreachable. Thus, more sophisticated neighborhood management is desirable, especially since this can have important implications for the topology visible to a routing protocol, and can substantially affect performance.

Channel Quality Estimation   Design of efficient probing strategies for channel estimation is an important direction for future work, with need for theoretical solutions, as well as practical strategies based on theoretical insight. Some results on optimal probing strategies for single user/link case are available in the literature, e.g., [17]. But there is dearth of approaches that take the multi-link, multi-hop setting into account. A related issue is that of reacting to a jammed or highly noisy channel.

Suitable Decision Policies   The LL maintains a wide range of statistics pertaining to traffic and channels. There is typically a different degree of confidence for different statistics (depending on the frequency of observation or reports). Thus, it would be desirable to adopt an approach in which the response to an observation is dependent on the degree of confidence, i.e., one could vary the degree of hysteresis based on degree of confidence (if more confident, the protocol can react more promptly; if less confident, the response can have more damping). Formulating such policies is an interesting direction for future work.

Implementing a Link Layer Reordering Buffer   Since the LL performs data-striping, there is a likelihood of out-of-order packet delivery, when nodes have multiple R-interfaces. This could be rectified by having a reordering buffer at the receiving transport endpoint.

However, it may be desirable to keep the LL's functions completely transparent to the higher layers, so that no changes to the higher layers are required for using the link layer protocol. Thus, it may be useful to implement a reordering buffer at the link-layer. Since the data-striping would performed by each local link-layer over each link, this can be done by using link layer sequence numbers for all transmitted packets over a link, and holding received out-of-order packets in a buffer till prior sequence numbers have been received. This can also enable TCP traffic to derive benefit from LL data-striping (our current simulations indicate that TCP does not benefit much).

Routing   In this chapter, we described a link layer protocol that performs dynamic adaptation. Given that this protocol address issues arising from heterogeneity of interfaces and channels at the link layer, it is of interest to devise a routing protocol that does not have any knowledge of specific low-level details of channels/radios, etc. This protocol would take an abstracted link/node cost metric from the LL, and use it for route-selection (with the route being a sequence of nodes). With such an approach, the same routing protocol can work in a diverse set of scenarios with different hardware specifications, since the knowledge of low-level details is encapulated by the LL.

Distance-vector routing is typically not very suitable for the envisioned scenarios, as it does not provide enough flexibility in quantifying the cost of a route. If proactive routing is desired, link-state routing appears to be the best fit. If reactive routing is desired, source-routing seems to be most appropriate. The key challenge lies in designing a suitable metric that is capable of capturing traffic-levels (which lead to interface bottlenecks), available channel/interface diversity along path, and long-term link conditions along the path. However, any traffic-based cost exposed to the routing layer should typically be computed over a longer timescale than costs used by the LL decisions, else instability may result [54]. Moreover, since the LL may locally adapt and cause channel-switching anytime during the lifetime of a route, the metric should typically not be based on current channel of operation of interfaces; rather it should take into account the channel-diversity available in the form of the channel-pool.

Extension to wider range of heterogeneous hardware capabilities   Another direction involves extending the envisioned stack architecture to address a wider range of

heterogeneous hardware capabilities, e.g., consider a scenario involving multiple heterogeneous radios/channels, as well as heterogeneous antennas. Such an effort can be quite useful, and can provide a generic design template for a wide range of scenarios.

Similarly, one could try to extend the scope to include making decisions about rate/power at the link layer, as well as address scenarios where two interfaces of the same type may have different number/type of antennas, yielding different reachability characteristics (whether and at what rate one can directly communicate with a nearby node). To an extent, the current design is capable of serving as a template for this wider range of scenarios. The current design assumes that reachability characteristics are solely a function of the channels that a node can be reached on; thus we have a set of channel queues. One could extend this to a set of queues for various combinations of choices (instead of a separate level of interfaces queues with a separate IF-scheduling, it would be reasonable to include the interface-choice as part of the combination); the CH-scheduler can still be used by defining appropriate conflict relations between these queues.

However, a major issue in addressing such multi-parameter adaptation is the resultant increase in unpredictability. In the currently addressed scenario, the reachability characteristics are a function of the channel, and of the availability of interfaces capable of switching on the particular channel at each node under consideration; they are largely, though not exclusively, determined by the R-channel selection which operates over much longer timescales than packet scheduling; furthermore packet scheduling decisions are done over a quantum of packets, while the net datarate estimate (used in the channel-selection decision) is updated after every packet). The greater the number of adaptable parameters, the greater is the dependence of the achievable rates on the decisions being made by other nearby nodes per packet, which increases complexity. To handle this, it may be beneficial in incorporate more structure in terms of potential multi-timescale parameter tuning (akin to the current channel restriction), as well as possibly increasing the scheduling quantum size (the goal being not amortization of overhead, but achieving predictability in what will happen over the timescale of next few packets).

Thus, there are many interesting directions worthy of exploration.

Chapter 7

Reliable Broadcast in Failure-prone Wireless Networks

The increasing use of wireless networks in critical application scenarios provides motivation for designing reliable communication algorithms that can leverage the distinct characteristics of the wireless channel. In this chapter, we introduce the reliable broadcast problem in the wireless context, and describe the underlying model and assumptions for the results in subsequent chapters. We also discuss related work.

7.1   Assumptions

We consider an idealized wireless network. There is a single common channel of operation, and all nodes are equipped with a single half-duplex transceiver. The wireless channel is assumed to be perfectly reliable, i.e., if a node transmits a message, and no other node in the vicinity is transmitting simultaneously (i.e., if no collisions occur), then the message is guaranteed to be received by all nodes within its range (termed its neighbors). Note that this idealized shared wireless channel intrinsically preserves ordering of messages sent by a node, i.e., if a node transmits messages m_1 and m_2 respectively in order, they will be received in that same order by all neighbors. We call this idealized behavior the *reliable local broadcast* assumption. While this assumption does not hold *per se* in real wireless networks, it may be possible to implement a local broadcast primitive that can provide probabilistic guarantees (given the probabilistic nature of wireless channel losses, a fully deterministic approach is not feasible in reality). Such a primitive could then be used as a subroutine by a global broadcast algorithm.

We assume *synchronous* communication. More specifically, for the results in Chapter 8 and Chapter 9, we assume that there is an underlying collision-free TDMA schedule, where time is divided into rounds, and each node has a designated transmission slot, which it can use to transmit without interfering or being interfered with, if it needs to. If a message is

173

transmitted by a node, then it is received by all its neighbors within a bounded amount of time (i.e., by the end of the slot).

Another assumption is that all nodes adhere to the collision-free schedule; even the faulty nodes do not deliberately cause collisions by transmitting out-of-turn. Similarly, they do not spoof the MAC addresses of other nodes. One way to view this situation is that the physical (PHY) and medium access control (MAC) layers of all nodes are fault-free, and the MAC layer does not allow higher layers to cause a change of MAC addresses. Thus, if all nodes have *a priori* unique MAC addresses, then each transmitted message (packet) will carry the true and unique identity of the node that transmitted the packet in its MAC header. Note that this means that each node knows the correct identity of the previous hop node from which it received the packet.[1] However, if the packet traversed multiple hops, the identity of the original sender or the previous hop relays (if included in the message contents), may be subject to tampering by a faulty relay.

For our results in subsequent chapters, we consider two distance metrics: L_∞ and L_2. The L_∞ metric is the metric induced by the L_∞ norm, such that the distance between points (x_1, y_1) and (x_2, y_2) is given by $\max\{|x_1 - x_2|, |y_1 - y_2|\}$ in the this metric.

The L_2 metric is induced by the L_2 norm, and is the Euclidean distance metric. The L_2 distance between points (x_1, y_1) and (x_2, y_2) is given by $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.

7.2 Problem Definition

The reliable broadcast problem for a designated source is defined as follows:

There is a designated source node in the network, which can originate a message for broadcast to the rest of the nodes in the network. The goal is to ensure that if the source is non-faulty, every non-faulty node in the network should correctly receive and determine the value originated by the source; if the source is faulty, all non-faulty node should agree on some common value. When a node decides upon some value as being the broadcast value, we say that it *commits* to it.

---

[1] The assumption that MAC addresses cannot be spoofed is also relevant to scenarios where link-layer authentication mechanisms are available, but end-to-end authentication is not. This is quite pertinent to sensor network deployments, where end-to-end authentication may involve too much overhead to be justifiable, but link-layer authentication may be feasible as it is much more lightweight. Link-layer authentication would assure that a node receiving a message is certain of the identity of the neighbor that transmitted that message.

7.2.1 Implications of Reliable Local Broadcast Assumption

As per the reliable local broadcast assumption, if a node transmits a message, all its neighbors are able to receive it, and are able to do so within a bounded amount of time. This greatly simplifies the task of achieving reliable broadcast in the presence of a faulty source node. Suppose the source is faulty. There are two ways in which it could manifest faulty behavior: (1) not send a message when other nodes expect it to do so, or (2) send two conflicting versions of the same message containing different values. If case (1) occurs, then neighbors of the source can use a simple timeout mechanism, whereby, if no message is received from the source within a certain interval of the expected time, they commit to a default value, and take the appropriate steps stipulated by the algorithm being followed to propagate it further. If case (2) occurs, all neighbors receive both values, and the duplicity of the source is detected. Thus the non-faulty neighbors of the source can again follow some default procedure (either commit to a default value, or to the first value received from the source), and take appropriate subsequent steps. Therefore, the source has no incentive to be duplicitous.

7.3 Related Work

We now review some existing work on reliable communication in the presence of faults.

Reliable communication under Byzantine failures has been studied for point-to-point communication networks under various assumptions [2]. The seminal result of Pease, Shostak and Lamport [89], [70] states that in case of full connectivity, Byzantine agreement with f faulty nodes is possible if and only if $n \geq 3f+1$. Under more general communication graphs, the requirements for Byzantine agreement are that $n \geq 3f + 1$, and the network be at least $(2f + 1)$-connected [26]. Byzantine agreement in k-cast channels has been considered in [21]. However this does not capture the spatially dependent connectivity that characterizes radio networks. Reliable broadcast in radio networks has also been studied in [60] and [57]. In [57], an infinite grid network was considered. A locally-bounded fault model was proposed, wherein an adversary was allowed to place faults subject to the constraint that no neighborhood have more than t faults. It was shown that under a Byzantine failure model, reliable broadcast is not achievable for $t \geq \lceil \frac{1}{2}r(2r + 1) \rceil$ (in both L_∞ and L_2 metrics). Besides a protocol was described that was able to achieve reliable broadcast

under the following conditions:

- If $t < \frac{1}{2}(r(r + \sqrt{\frac{r}{2}} + 1))$, then reliable broadcast is achievable in the L_∞ metric.

- If $t < \frac{1}{4}(r(r + \sqrt{\frac{r}{2}} + 1)) - 2$, then reliable broadcast is achievable in the L_2 metric.

This protocol stipulates that nodes wait till they hear the same value from $t + 1$ neighbors before they commit to it, and re-broadcast it exactly once for the benefit of other neighbors. Under this protocol, no non-faulty node will ever accept the wrong value. However, there is a possibility of some nodes never being able to decide, and the achievability bounds do not match the impossibility bound, leaving a region of uncertainty.In [112], a tight bound for tolerable t using the simple broadcast protocol of [57] was established.

Further study of the locally bounded fault model is undertaken in [90], where arbitrary graphs are considered instead of a specific network model. While the discussion mentions both radio and message-passing networks, there is an assumption that duplicity by the source (sending different messages to different neighbors) is impossible. Upper and lower bounds for achievability of reliable broadcast are presented, based on graph-theoretic parameters, for arbitrary graphs. However, no exact thresholds are established. Two broadcast algorithms are considered. One is the simple algorithm of [57] that is referred to as the Certified Propagation Algorithm (CPA). Another algorithm, termed as the Relaxed Propagation Algorithm (RPA), is described, which is t-locally safe (i.e., no non-faulty node will commit to an incorrect value by following it). It is shown that RPA is a more powerful algorithm, as there exist graphs for which RPA succeeds but CPA does not. It is also shown that there exist certain graphs in which algorithms that work with knowledge of topology succeed in achieving reliable broadcast, while those that lack this knowledge fail to do so. The RPA algorithm and our algorithms for reliable broadcast described in Chapter 8 are quite similar, as there is a reliance on receiving indirect reports about values committed to by nodes through a sufficient number of node-disjoint paths.

Scenarios involving a collision-causing adversary are addressed in [58, 38, 27]. The issue of achieving broadcast when a (locally bounded) adversary can cause bounded a bounded number of collisions or address spoofing is handled in [58]. It presents protocol transformations that can lead to resilience to a bounded number of collisions or address spoofing attempts. It uses the protocol described in Section 8.4 of Chapter 8 as a building block. However the result is based on the assumption that non-faulty nodes are not hindered

by energy-limitations, and can retransmit messages as many times as needed. The impact of an energy-budget on consensus has been studied for a single-hop setting in [38], and it has been proved that non-faulty nodes would require at least incrementally larger budget than faulty nodes to arrive at a consensus. In [91], conditions for broadcast have been established under a probabilistic transient failure model, where faulty behavior also includes the possibility of nodes causing collision.

Probabilistic failure are considered in [91] which examines the case of message-passing and radio networks with random transient failures. The transient failure behavior includes the possibility of causing collision.

Communication of information in a single-hop multi-channel wireless network with a malicious adversary that can cause collisions concurrently in a limited number of channels has been considered in [27].

Also related is work in [109] on unknown fixed identity networks; this work assumes that nodes cannot fake their identity to their neighbors. Our model also has a similar assumption.

7.3.1 Crash-stop Failures

For crash-stop faults, the reliable broadcast problem reduces to the connectivity problem.

Crash-stop failures are considered in [60] for finite networks comprising nodes located in a regular grid pattern. The focus is on obtaining algorithms for efficient broadcast to the part of the network that is reachable from the source, and not on quantifying the number of faults that render some nodes unreachable.

A grid network model was considered in [100] where nodes are located at integer lattice sites on a square grid, and fail independently. Nodes have a common transmission range r. The probability of not failing is specified as p, and it is shown that a sufficient condition for connectivity and coverage is that transmission range r must be set to ensure that node degree is $c_1(\frac{\log n}{p})$ (for some constant c_1). It is also shown that a necessary condition for coverage (and hence for joint coverage and connectivity) is that node degree be at least $c_2(\frac{\log n}{p})$ (for another constant c_2. A fallacy in the above necessary condition was pointed out by [62], and a subsequent correction [102] by the authors of [100] presents examples illustrating that the necessary condition may fail to hold for certain sub-ranges of

p. The issue of coverage has been examined in detail in [62] for random, grid, and Poisson deployments. However, the necessary and sufficient conditions formulated by them take a more complex form, and do not point to a single $f(n, p)$ such that a degree of $\Theta(f(n, p))$ is both necessary and sufficient for asymptotic coverage. Besides, the necessary condition is formulated for the specific case when $\lim_{n \to \infty} p \to 0$.

We have also derived results for crash-stop failures in a grid network that yield a different expression than [100], and while our results are within a constant factor of their results for most values of p, our results are more accurate when $p \to 0$.

In [42], it was proved that in a unit area network with uniformly distributed node placement, where nodes have a common transmission radius r, such that $\pi r^2 = \frac{(\log n + c(n))}{n}$, the network is asymptotically connected with probability one iff $c(n) \to \infty$. This constitutes the case $p = 0$ for random networks. Recently, necessary and sufficient conditions for asymptotic connectivity in a random network with low duty cycle sensors have been formulated in [55]. This is equivalent to the problem of crash-stop failures in a random network.

On a related note, fault-tolerant consensus (in the presence of channel unreliability and crash-stop failures) has been studied in [20]. The focus is primarily on a single-hop network, though some simulation results for a multi-hop setting are also reported.

7.3.2 Reliable Local Broadcast

Much of the theoretical work mentioned earlier assumes that the wireless channel itself is perfectly reliable. The lossy nature of the channel is not accounted for, and thus many of these results are not directly applicable to a real-world scenario. A proposal to reconcile the theory and practice of wireless broadcast has been made in [19]. They identify certain properties that a reliable local broadcast should have. They introduce some models to capture the nature of losses and collisions, viz., the No-Collisions(NC) model, the Eventual No-Collisions (ENC) model, the Total Collision (TC) model, and the Partial Collision (PC) Model. In a single-hop network conforming to the TC-model, it is shown that consensus is achievable with any number of Byzantine/crash-stop failures. However, practical realization of the TC model is not delved into in detail (though some possibilities are hinted at).

Another relevant body of work pertains to reliable multicast with probabilistic guarantees [13], [78] which seeks to achieve a scalable solution with probabilistic guarantees.

7.3.3 Fault Detection

A related area pertains to failure detection. Algorithms that detect failure can be very useful, as messages received from nodes detected as faulty can then be excluded from future communication. This can help improve efficiency. A seminal work in the area of failure detection is the PMC Model [94] proposed by Preparata, Metze and Chien. [16] also pertains to this theme. Results for failure-detection in a scenario with locally bounded faults are described in [68]. This work is quite relevant as the locally bounded model is also addressed by us in Chapter 8, in the context of reliable broadcast. Self-adjusting Byzantine Agreement is considered in [129]. This work describes how the Byzantine nodes can be progressively detected in a network; at most a certain number of broadcast instances can fail before all faults get detected.

Chapter 8

Reliable Broadcast with Locally Bounded Failures

In this chapter, we study the reliable broadcast problem with a *locally bounded* fault occurence model, which was briefly introduced in Chapter 7. We begin by describing the model and notation in Section 8.1, and then summarize the chapter results in Section 8.2. We formulate a sufficient condition for achieving reliable broadcast in a general graph with such a fault model in Section 8.3. In Section 8.4, we establish a bound for achievability of reliable broadcast in a grid network model for the Ł$_\infty$ metric. This bound matches an impossibility bound proved in [57], and thus establishes the exact threshold for this model. In Section 8.6, we describe an approximate result for the Ł$_2$ (Euclidean) metric. We describe an alternative broadcast algorithm in Section 8.7 which is also optimal in the grid network for the L_∞ metric, in the sense that it can tolerate the maximum number of tolerable faults. We discuss interesting issues and future directions in Sections 8.8 and 8.9 respectively.

8.1 Preliminaries

We consider an infinite wireless network, with nodes situated on a grid (where each grid square has side 1), under Byzantine and crash-stop failures. Note that the grid defines the spatial layout of nodes, and not the network topology. All nodes use a common transmission range r, which is assumed to be an integer. As described in Chapter 7, two distance metrics, L_∞ and L_2, are considered. In the L_∞ metric, each node has exactly $4r^2 + 4r$ neighbors. The results also hold for a finite toroidal network in which r is smaller than the network radius. In scenarios where the entire network region is within distance r of the designated source, reliable broadcast is trivially always achievable due to the reliable local broadcast assumption.

In the grid network, nodes are identified by their grid location i.e. (x, y) denotes the node at (x, y). The neighborhood of (x, y) comprises all nodes within distance r of (x, y)

(according to the distance metric in consideration) and is denoted as $nbd(x, y)$. For succint description, we define a term $pnbd(x, y)$ where $pnbd(x, y) = nbd(x - 1, y) \cup nbd(x + 1, y) \cup nbd(x, y - 1) \cup nbd(x, y + 1)$. Intuitively $pnbd(x, y)$ denotes the *perturbed neighborhood* of (x, y), obtained by perturbing the center of the neighborhood to one of the nodes at unit distance from (x, y) on the grid.

A non-faulty node shall be variously alluded to as an non-faulty or correct node, while a node exhibiting Byzantine failure shall occasionally be referred to as a malicious node. We shall occasionally refer to $nbd(S)$ where S is a set. In such cases, $nbd(S) = \bigcup_{x \in S} nbd(x)$.

The locally-bounded fault occurrence model is considered, wherein an adversary is allowed to place faults as it chooses, so long as no single neighborhood contains more than t faults. When we refer to the neighborhood of a node v, it includes v itself. Thus a correct node may have up to t faulty neighbors, while a faulty node may have up to $(t - 1)$ neighbors that are also faulty.

As was discussed in Chapter 7, we assume that the a node may not spoof another node's MAC address, and resultantly, any node knows the correct identity of the previous hop node from which it received a message. No collisions are possible, i.e., there exists a pre-determined collision-free TDMA schedule that all nodes follow.

A designated source (that is assumed to be located at the origin of the grid coordinate system, w.l.o.g.) broadcasts a message with a binary value. The objective is to ensure reliable broadcast of this value (see the definition of the reliable broadcast problem in Section 7.2).

8.2 Summary of Results

We prove the following results:

1. We describe a general sufficient condition for reliable broadcast in a general network graph under the reliable local broadcast assumption, which provides intuition for the subsequent grid network results.

2. We present a lower bound in L_∞ metric on the maximum number of Byzantine failures t that may occur in any given neighborhood without rendering reliable broadcast impossible in the grid network model. We provide a constructive proof by describing

two algorithms that both achieve reliable broadcast in the L_∞ metric whenever $t < \frac{1}{2}r(2r+1)$. This exactly matches an impossibility bound proved in [57], and thus establishes an exact threshold for Byzantine agreement under this network model. For completeness, we also study crash-stop failures, and prove that reliable broadcast is achievable with locally bounded crash-stop failures iff the number of faulty nodes in any neighborhood is $t < r(2r+1)$ (in the L_∞ metric).

3. We present approximate bounds for L_2, i.e., Euclidean metric, and show that when r is sufficiently large, the thresholds must lie in a similar range as $L\infty$. In particular, we argue that for sufficiently large r, Byzantine agreement is indeed possible in Euclidean metric if slightly less than one-fourth of the nodes in any given neighborhood may be faulty, while it is possible to tolerate crash-stop failures if they are slightly less than half the neighborhood population.

A preliminary version of some of the chapter results was reported in [5].

8.3 A General Sufficient Condition

Consider a general undirected graph $G = (V, E)$, whose topology is known to all network nodes. Designate a source $s \in V$ as the source of the broadcast. A s-cut is a partition $C = (S, V \setminus S)$ such that $s \in S$. In the course of a broadcast operation, S can potentially denote the set of nodes that have already had the opportunity to correctly determine the broadcast value, and commit to it (note that all non-faulty nodes in S will thus indeed have committed to the correct value, while the behavior of faulty nodes is indeterminate). $V \setminus S$ can potentially denote the set of nodes that are yet to do so.

Let us consider the case where G is a finite graph. In this case, any cut C may be considered as an envelope for the advancing frontier of the broadcast at some instant, with further expansion of the frontier depending on the existence of sufficient connectivity across the cut. If the cut C were indeed encountered during algorithm operation, this is evidently true. However, even if the cut $C = (S, V \setminus S)$ were not actually encountered during algorithm operation, the following argument can be made:

At any point of time t during algorithm execution, let the actual frontier be denoted by the cut $C_{actual}(t) = (S_{actual}(t), V \setminus S_{actual}(t))$. Consider an algorithm step at time t' such
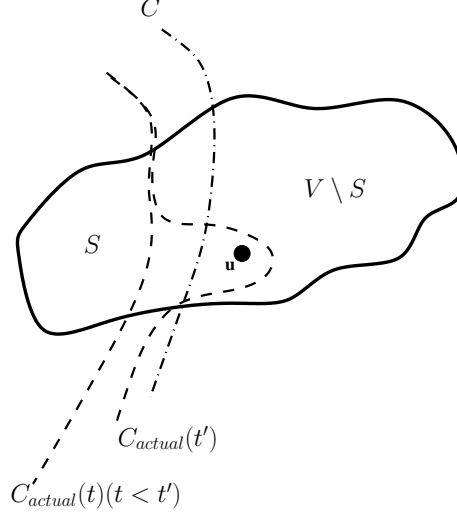
Figure 8.1: Equivalence of Cut Conditions

that for all $t < t'$, $S_{actual}(t) \subseteq S$, but $S_{actual}(t') \nsubseteq S$. Thus, at time t', at least one node $u \in V \setminus S$ crossed over from $V \setminus S$ to S (i.e., received sufficient information to be able to commit to the correct value, and, if it is non-faulty, indeed committed to it) from $V \setminus S_{actual}$ to S_{actual}. At time $t < t'$, the frontier of the broadcast (i.e., C_{actual}) lay strictly behind the frontier defined by $C = (S, V \setminus S)$. Thus, if a node has access to sufficient information flowing to it from S_{actual} to be able to cross-over, then it must necessarily have access to at least as much information flowing to it from S (since the network topology, and hence paths in the network, are the same in both cases, and the set of nodes that already definitively know the correct value in the latter case is a superset of that in the former case), and be able to cross the cut $C = (S, V \setminus S)$, if it had been encountered. This is depicted in Fig. 8.1. Hence, the following two statements are equivalent:

- *Statement 1:* For every s-cut $(S, V \setminus S)$ of the graph that is actually encountered during algorithm execution, some node $u \in V \setminus S$ possesses sufficient connectivity to be able to cross over to S from $V \setminus S$.

- *Statement 2:* For every possible s-cut $(S, V \setminus S)$ of the graph, assuming all nodes in S have had the opportunity to make a correct determination (and non-faulty nodes have actually made it), some node $u \in V \setminus S$ possesses sufficient connectivity to be able to cross-over to S.

Hence, for a finite graph, Statement 2 does not impose a more stringent requirement

183

than Statement 1. We remark that the use of the notation t for time in the prior discussion should not be confused with the subsequent use of t to denote the maximum number of faults in any single neighborhood.

Lemma 43. *Given a finite undirected graph $G = (V, E)$, Statement 1 is a sufficient condition for feasibility of broadcast, and Statement 2 is an equivalent sufficient condition.*

Proof. This may be seen as follows: since Statement 1 holds for every encountered cut, the set $V \setminus S$ will continue to decrease, and being finite will eventually become empty. At that stage $S = V$, and the broadcast will have successfully reached every node (and all the non-faulty nodes will have made a determination of the correct value). Statement 2 is equivalent to Statement 1, and is hence also a sufficient condition. $\square$

It now remains to characterize what constitutes sufficient connectivity to be able to cross over to the source side of the cut. The goal of any reliable broadcast algorithm is that each non-faulty node should be able to eventually decide on the correct broadcast value. If at any instant, the frontier is represented by cut $C = (S, V \setminus S)$, then by the assumption of Statement 2, all nodes in S have correctly determined the broadcast value. Any communication of information across the cut must happen through the nodes in $C_S = \{v \in S | \exists (v, u) \in E \text{ such that } u \in V \setminus S\}$. Therefore, for the purpose of analysis, it suffices to transform the source side of the cut S to $S' = s_{sup} \cup C_S \cup (nbd(C_S) \cap S)$, with s_{sup} being a new *super-source* node that acts as an abstract sender of the correct broadcast value, and is connected directly to each node in C_S (via pseudo-edges).[1] Other edges between included vertices are preserved. The neighbors of vertices in C_S on the source side are included to enforce the per-neighborhood fault constraint amongst the vertices in C_S. We refer to the corresponding graph induced by $V' = S' \cup (V \setminus S)$, with the pseudo-edges added, as the reduced graph $G' = (V', E')$.

We state and prove the following sufficient condition:

Theorem 14. *Given a finite undirected graph $G = (V, E)$ and designated source s, with upto t byzantine faults in any neighborhood, reliable broadcast is achievable in G if every s-cut $C = (S, V \setminus S)$ (with C_S denoting the set of vertices that have at least one incident edge*

---

[1] This captures the fact that all non-faulty nodes in C_S have determined the correct value.
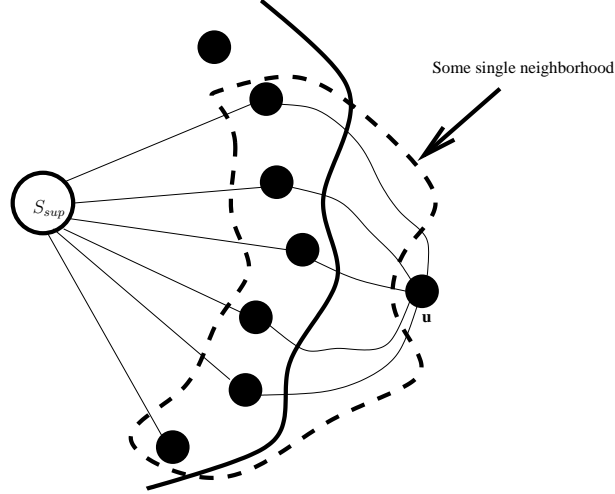
Figure 8.2: Connectivity to super-source

crossing the cut) satisfies the following: $\exists\, u \in V \setminus S$ such that either $(s, u) \in E$ or there exist $(2t+1)$ node-disjoint $s_{sup} \rightsquigarrow u$ paths in the transformed graph G', such that all intermediate nodes on these paths lie within the neighborhood of some single node $v \neq s_{sup} \in V'$.

Proof. Since all nodes in S, and hence $C_S \subseteq S$, have had the opportunity to correctly determine the broadcast value (by assumption), the addition of pseudo-edges with s_{sup} ensures this same property (since neighbors of the source can trivially determine the value correctly due to the reliable local broadcast assumption), while removing from consideration nodes that are no longer relevant to the result we seek to prove. If a node is connected to s_{sup} via at least $2t + 1$ node-disjoint paths that all lie within some single neighborhood, then at most t of these paths may have a faulty node (as no more than t faults may exist in any single neighborhood).[2] Thus, the node u will eventually receive the correct value over at least $t + 1$ node-disjoint paths, and will be in a position to commit to it. The situation is illustrated in Fig. 8.2.

By Lemma 43, this is a sufficient condition for finite graphs. $\qquad\square$

Corollary 5. *Given a finite undirected graph $G = (V, E)$ and designated source s, with upto t crash-stop faults in any neighborhood, reliable broadcast is achievable in G if every*

---

[2]Also note that each node is aware of the correct identity of the previous hop node from which it received a message, and thus the identity of the last faulty node on a path is always revealed; hence u will not consider any other path through this faulty node when counting the number of disjoint paths through which a value was received. This ensures that u will count at most t faulty paths for a value, and prevents faulty nodes from confusing u even if they tamper with previous hop path information.

s-cut $C = (S, V \setminus S)$ *(with* C_S *denoting the vertices for which at least one incident edge crosses the cut) satisfies the following:* $\exists u \in V \setminus S$ *such that either* $(s, u) \in E$ *or there exist* $(t+1)$ *node-disjoint* (s_{sup}, u) *paths in the reduced graph* G', *such that all intermediate nodes on these paths lie within the neighborhood of some node* $v \neq s_{sup}$.

Proof. When crash-stop failures are considered, reachability is synonymous with achievability of reliable broadcast. If a non-faulty node is a neighbor of s, it will trivially receive the broadcast. If a node is connected to $t + 1$ nodes in S via one path each such that all $t + 1$ paths are node-disjoint, and lie in a single neighborhood, then at most t of these can be faulty. Thus, there will be at least one fault-free path through which the node may be reached, and the broadcast can propagate further. □

Infinite Graphs For any finite fault-threshold t, one can argue that Theorem 14 also holds for infinite graphs as follows: Suppose the condition stated in Theorem 14 holds, but it is impossible for some nodes to determine the correct broadcast value. Consider the set D comprising all such nodes that are not able to eventually determine the correct value. Evidently, none of the nodes in D can be a neighbor of the source s, else such a node would trivially have the opportunity to determine the correct value. Therefore, they are all non-neighbors of s. By assumption, all nodes in $V \setminus D$ eventually have the opportunity to determine the correct value. Consider the corresponding cut $(V \setminus D, D)$. Then, using the proof argument of Theorem 14, there exists some node $u \in D$ such that there are at least $2t + 1$ node-disjoint $s_{sup} \rightsquigarrow u$ paths in the transformed graph G' for cut $(V \setminus D, D)$. Consider exactly $2t + 1$ of these paths. Note that the nodes neighboring s_{sup} on these paths are $2t + 1$ in number and belong to $V \setminus D$. By assumption, in the actual network, these $2t + 1$ nodes will eventually have the opportunity to determine the correct value. Once these $2t + 1$ nodes have had the opportunity to determine the correct value, u would also eventually receive information from enough node-disjoint paths, and have the opportunity to determine the correct value. This yields a contradiction.

The same argument can be used for Corollary 5.

8.4 Byzantine Failures in a Grid Network

We prove the following result for locally bounded Byzantine failures in the grid network:

Theorem 15. *If $t < \frac{1}{2}r(2r+1)$, reliable broadcast is achievable in the grid network for the L_∞ metric.*

We present an algorithm to achieve reliable broadcast, based on the same intuition as the general sufficient condition of Theorem 14. Without loss of generality, we assume that the message comprises a binary value (say 0 or 1). A non-faulty node that is not the source is said to *commit* to a value when it decides that it is indeed the value originated by the source. The algorithm requires maintenance of state by each node pertaining to messages received from nodes within its two-hop neighborhood. The algorithm operates as follows:

- Initially, the source does a local broadcast of the message.

- Each neighbor i of the source commits to the first value v it heard from the source and does a one-time local broadcast of a $COMMITTED(i,v)$ message.

- Hereafter, the following algorithm is followed by each node j (including those involved in the previous two steps):

  On receipt of a $COMMITTED(i,v)$ message from neighbor i, record the message, and broadcast a $HEARD(j,i,v)$ message.

  On receipt of a $HEARD(k,i,v)$ message from neighbor k, record the message, but do not re-propagate.

  On committing to a value v, do a one-time local broadcast of a $COMMITTED(j,v)$ message.

  A node j commits to a value v, if it has not already committed to a value, and it becomes *certain* about value v. A node is said to be *certain* about a value v if it receives v through $COMMITTED$ or $HEARD$ messages over at least $t+1$ node-disjoint paths that lie within a single neighborhood. More precisely, a node j is *certain* of a value v if there is a node Q such that j received some $t+1$ messages $m_1, m_2, ..., m_{t+1}$ where $m_i = COMMITTED(A_i, v)$ or $m_i = HEARD(A_i, A_{i'}, v)$, and all the A_i, A_i' are *distinct* nodes lying in the neighborhood of Q.[3]

---

[3]A faulty intermediate node can alter the affixed identity of the previous node listed in the $HEARD$ message (this is part of the message content, which can be altered). This does not cause a problem as the identity of such a faulty intermediate node (let us call it x) on the forwarding path will always be revealed

Theorem 16. *No non-faulty node shall commit to a wrong value by following the previously described algorithm.*

Proof. The proof is by contradiction. Consider the first non-faulty node, say j, that makes a wrong decision to commit to value v. Evidently, j cannot be a neighbor of the source. This implies it received the value v from at least $t+1$ nodes through a single path (direct or two-hop) each, such that all $t+1$ paths are node-disjoint, and lie in some single neighborhood. Since the number of faults in any single neighborhood may be at most t, it implies that at most t of these paths could have a faulty source (of a *COMMITTED* message) or a faulty intermediate node (that sends a *HEARD* message). Thus, all paths cannot have relayed the wrong value, and so v must indeed be the correct value. □

Theorem 17. *Each non-faulty node is eventually able to commit to the correct value.*

Proof. We prove that each non-faulty node will be able to meet the conditions stipulated by the algorithm for committing to the correct value. The proof also clarifies the operation of the algorithm. Intuitively, the essence of the proof lies in showing that each node P (other than the direct neighbors of $(0,0)$ which can trivially determine the correct value) can receive information from a part of the network that has already committed to the correct value, along $(2t+1)$ node-disjoint paths lying in some single neighborhood. This is akin to the general sufficient condition of Section 8.3.

The proof is by induction.

Base Case: All non-faulty nodes in $nbd(0,0)$ are able to commit to the correct value. This follows trivially from our assumed model since they all hear the source directly.

Inductive Hypothesis: If all non-faulty neighbors of a node located at (a,b) i.e. all non-faulty nodes in $nbd(a,b)$ are able to commit to the correct value, then all non-faulty nodes in $pnbd(a,b)$ are able to commit to the correct value.

---

to j (x must j's neighbor, as the forwarding paths involve only two hops, and hence j knows its identity as MAC addresses cannot be spoofed). Resultantly, even if x has altered the identity of the node before it on a forwarding path, this is acceptable, as j will not include any other message with a path through x in the set of $t+1$ messages, and resultantly given only t faulty nodes in the neighborhood, at most t out of the $t+1$ paths can involve faulty information.

Proof of Inductive Hypothesis: We show that for each node P in $pnbd(a,b) \setminus nbd(a,b)$ there exists a set of 2t+1 paths $\{\pi_1, \pi_2, ..., \pi_{2t+1}\}$ of the form $\pi_i = (A_i, P)$ or $\pi_i = (A_i, A'_i, P)$, such that all A_i, A'_i are distinct, lie in some single neighborhood, and all $A_i \in nbd(a,b)$. Since no more than t of the A_i, A'_i can be faulty, this guarantees that the node will receive the correct value through at least $(t+1)$ paths, and will also commit to it.

Consider a node P belonging to $nbd(a, b+1)$. The argument for nodes in $nbd(a, b-1), nbd(a-1, b)$ and $nbd(a+1, b)$ is similar.

Node P in $nbd(a, b+1) \setminus nbd(a, b)$ may be considered to be located at $(a-r+p, b+r+1)$ where $\{0 \le p \le 2r\}$ (Fig. 8.3). We present an explicit argument for locations of P corresponding to $\{0 \le p \le r\}$. A similar argument holds for the remaining locations, by virtue of symmetry.

We show the existence of $r(2r+1)$ node-disjoint paths $\pi_1, \pi_2, ..., \pi_{r(2r+1)}$, that all lie within the same single neighborhood (centered at $(a, b+r+1)$, and indicated by the dark-edged square in Fig. 8.3). The region marked A comprises $\{(x,y)|(a-r) \le x \le (a+p); (b+1) \le y \le (b+r)\}$, and nodes in this region lie in $nbd(a,b)$, and are also neighbors of P. Thus, there are $r(r+p+1)$ paths of the form $A \to P$. The region B comprises $\{(x,y)|(a+p+1) \le x \le (a+r); (b+1) \le y \le (b+r)\}$, and falls in $nbd(a,b)$. The region B' is obtained by a translation of B to the left by r units, and then up by r units. Thus, region B' comprises $\{(x,y)|(a+p+1-r) \le x \le a; (b+r+1) \le y \le (b+2r)\}$, and falls in $nbd(P)$. Consequently, there is a one-to-one correpondence between a point (x,y) in B and a point $(x-r, y+r)$ in B', such that the points in each pair are neighbors. This yields $r(r-p)$ paths of the form $B \to B' \to P$.

Thus, $r(2r+1)$ node-disjoint paths are obtained.

Observe that the inductive hypothesis along with the base case suffice to show that every non-faulty node will eventually commit to the correct value, since starting at $(0,0)$, one can cover the entire infinite grid by moving up, down, left and right. Therefore, non-faulty nodes in the neighborhood of every grid point can be shown to be eventually able to determine the broadcast value. □
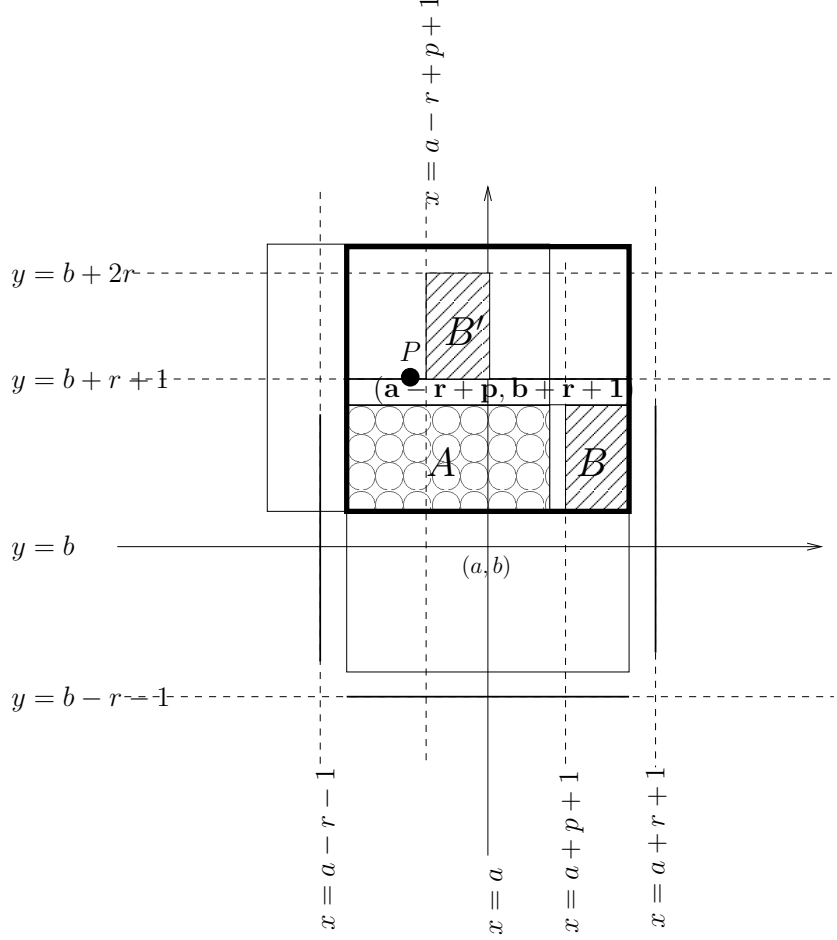
Figure 8.3: Existence of Sufficient Connectivity

8.5   Crash-Stop Failures in a Grid Network

When only crash-stop failures occur, the sole criterion for achievability is reachability, and no special algorithm is required. Each node that receives a value commits to it, re-broadcasts it once for the benefit of others, and then may terminate local execution of the algorithm.

In this failure mode, we establish an exact threshold for tolerable faults in L_∞ metric. The impossibility bound is trivial to derive but we state and prove it here for the sake of completeness.

Theorem 18. *Under a crash-stop failure model, if $t \geq r(2r+1)$, it is impossible to achieve reliable broadcast in the grid network, with the L_∞ metric.*

Proof. We present a construction with $t = r(2r+1)$ that renders reliable broadcast impossible. Consider the network in Fig. 8.4. The nodes in the designated region $\{(x, y) | a \leq$

$(0, 0)$

$x = a$ $x = a + r - 1$

Figure 8.4: Network Partition due to Crash Stop Failures

$x < a + r$} (for some $a \geq 1$) are all faulty while all other nodes are non-faulty. As may be seen, the maximum number of faulty nodes in any given neighborhood is at most $r(2r + 1)$. However this configuration partitions all nodes in the half-plane $x \geq a + r$ from the source and they are unable to receive the broadcast. □
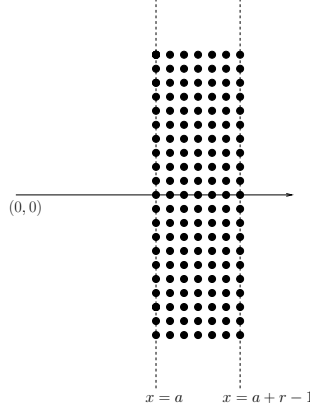
The achievability bound can be obtained from the result for the Byzantine model.

Theorem 19. *Under a crash-stop failure model, if $t < r(2r + 1)$, it is possible to achieve reliable broadcast in the grid network, with the L_∞ metric.*

Proof. Consider the proof for the byzantine fault-tolerant algorithm in Section 8.4. Given that $nbd(a, b)$ has decided, there exist $r(2r + 1)$ node-disjoint paths of the form described in Theorem 17 that lie in one single neighborhood. Since $t < r(2r + 1)$, at least one path will be fault-free, thereby enabling the broadcast to propagate to $pnbd(a, b)$. Thus, by inductive reasoning, all fault-free nodes on the grid will receive the broadcast. □

8.6   Euclidean Metric

In this section, we consider the issue of reliable broadcast in the L_2, i.e., Euclidean metric. We refrain from establishing exact thresholds as it is difficult to precisely determine lattice points falling in areas bounded by circular arcs. We present an *approximate* argument showing that reliable broadcast in L_2 is achievable if slightly less that one-fourth fraction of nodes in any neighborhood exhibit Byzantine faults. We work with the value $t < 0.24\pi r^2$.

191
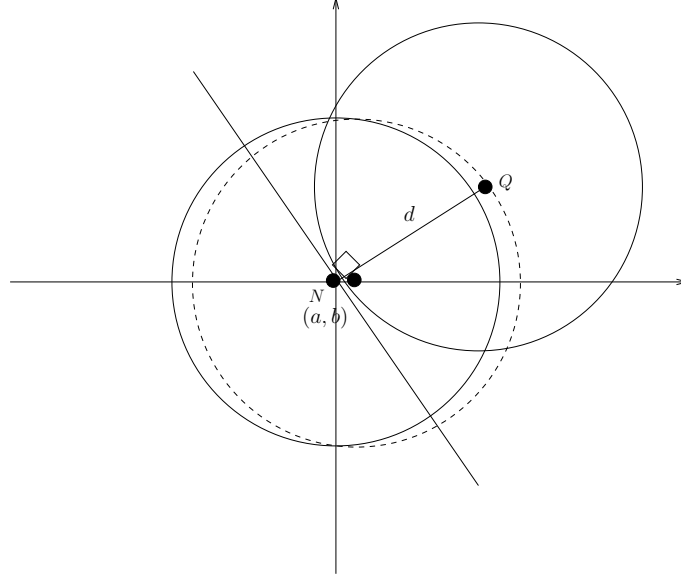
Figure 8.5: Illustrating an Approximate Argument for Euclidean Metric

The basis for the approximate argument is that, given a closed simple region of area A, and perimeter p, bounded by upto k straight line segments and circular arcs of radius r, where k is a small constant, the number of lattice point lying within it, N_l, is given by $N_l = A \pm O(p)$, and the constant hidden in the $O(p)$ term is small. The justification for this claim is based on Pick's Theorem [113], and is presented in Appendix D.

Therefore, for sufficiently large r, the number of nodes that lie in various considered subregions of a circle of radius r (elaborated later) are approximately $A \pm O(r)$ each (where A is the area of that subregion). Thus, we expect the argument to hold well for large values of r.

The argument uses induction, as in the previous section.

Base Case: All non-faulty nodes in $nbd(0,0)$ are able to commit to the correct value. This follows trivially since they hear the origin directly.

Inductive Hypothesis: If all non-faulty neighbors of a node located at (a, b) are able to commit to the correct value, then all non-faulty nodes in $pnbd(a, b)$ are able to commit to the correct value.
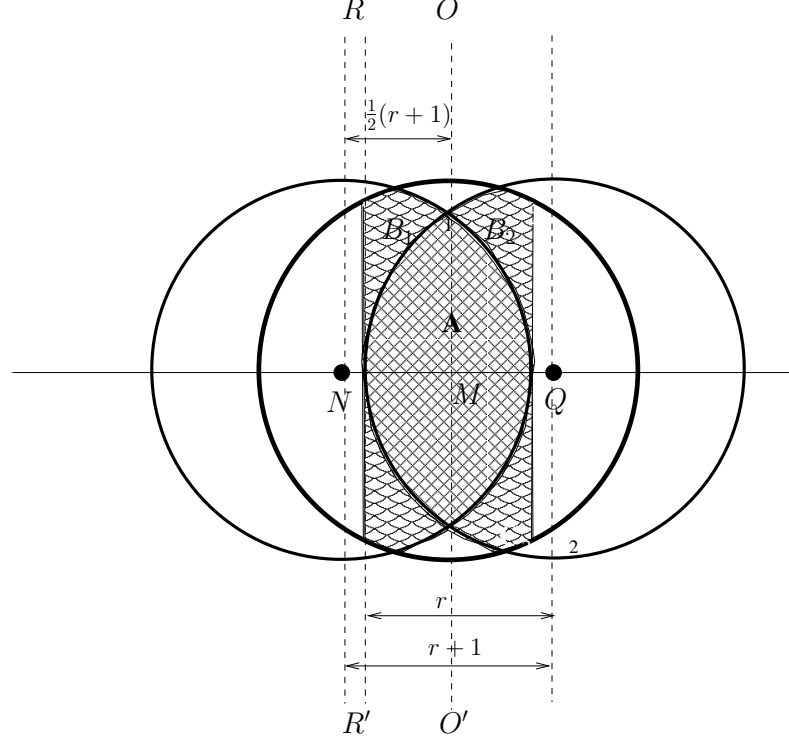
Figure 8.6: Approximate Construction depicting Node-Disjoint Paths (NQ from Fig. 8.5 rotated to x-axis)

Justification of Inductive Hypothesis: We show that each node in $pnbd(a,b) \backslash nbd(a,b)$ is connected to $2t+1$ nodes in $nbd(a,b)$ via one path each, such that all these $2t+1$ paths are node-disjoint and they all (the endpoints, as well as any intermediate nodes) lie entirely in one single neighborhood. Since no more than t of these can be faulty, this would guarantee that the node will receive the correct value through at least $t+1$ such paths, and commit to it.

Consider the node at (a,b), as in Fig. 8.5. Let d be the distance between the node at (a,b) (we call it node N) and any node in $(pnbd(a,b) \backslash nbd(a,b))$ (we call it node Q). Then $d \leq r+1$ (from the triangle inequality). It suffices to consider the possibility $d = r+1$, as that yields the least overlap between the neighborhoods of N and Q.

We consider the situation in Fig. 8.6 with NQ from Fig. 8.5 rotated to the horizontal axis for clarity of presentation. We attempt to construct node-disjoint paths that all lie within the neighborhood centred at M (the midpoint of NQ) or the grid location nearest to it. If M is itself not a grid point, the resultant perturbation of the neighborhood centre to the nearest grid location can only affect the presented calculations by $O(r)$. The set of

193

nodes marked A are common neighbors of P and Q and constitute one-hop paths $(A \rightarrow Q)$. A set of two-hop paths $B_1 \rightarrow B_2 \rightarrow Q$ is also formed where each point (x, y) in region B_1 has a corresponding point in B_2 (its image under reflection by axis OO'). Thus, in an approximate sense, for almost each grid-points in B_1, we can find a unique grid-point in B_2 with which it can be paired (with upto $O(r)$ unpaired grid points remaining).

The number of paths is thus approximately equal to the sum of the areas A and B_1, which turns out to be approximately $1.538r^2 = 0.49\pi r^2 \geq (2(0.24\pi r^2) + 1)$ for sufficiently large r. The details of the calculation are presented in Appendix D. Thus approximately $0.24\pi r^2$ Byzantine faults may be tolerated.

We also argue that reliable broadcast is not possible if $t \geq 0.3\pi r^2$ (approximately). The argument is based on a construction identical to that presented in [57] for L_∞, which is depicted in Fig. 8.7. As proved in [57], this arrangement of faults renders reliable broadcast impossible (see [57] for details). Note that the maximum number of faults lying in any single neighborhood is given by the number of faulty nodes in the circled region (Fig. 8.7). The relevant area is approximately $0.6\pi r^2$, and we expect approximately $0.6\pi r^2 \pm O(r)$ nodes to lie in it. Half of these, i.e., around $0.3\pi r^2 \pm O(r)$ are to be faulty. This yields the argument that if $t \geq 0.3\pi r^2$ (approximately), reliable broadcast would be unachievable. Thus the critical threshold for L_2 metric would lie between a 0.24 and a 0.3 fraction, i.e., in the vicinity of a one-fourth fraction of faults.

Observe that the above argument also implies that around $2t = 0.48\pi r^2$ crash-stop failures may be tolerated, while around $0.6\pi r^2$ failures per neighborhood would render reliable broadcast impossible.

8.7 An Alternative Broadcast Algorithm

In this section, we describe an alternative algorithm. Though this algorithm requires greater message overhead than the algorithm described in Section 8.4, it is of some interest, as it demonstrates the existence of a stronger localized connectivity property in the grid, which may possibly have relevance in contexts other than reliable broadcast.

As in Section 8.4, we assume w.l.o.g. that the message to comprise a binary value (say 0 or 1). A node that is not the source is said to *commit* to a value when it decides that it is indeed the value originated by the source.
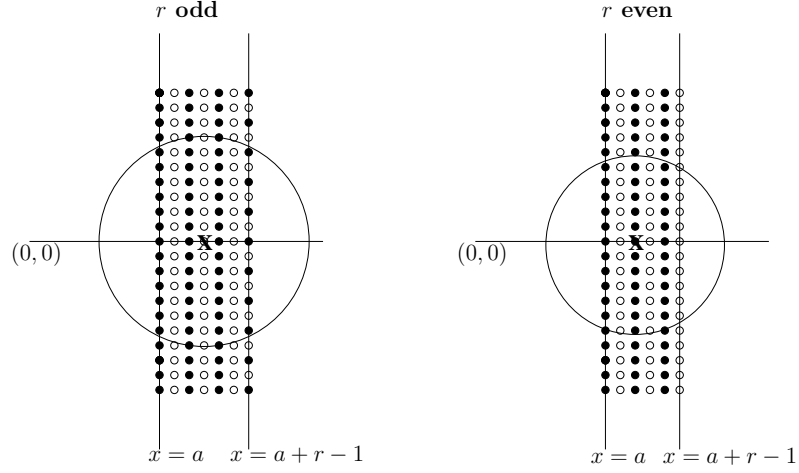
Figure 8.7: Impossibility Construction for Byzantine Failures in Euclidean metric

The algorithm requires maintenance of state by each node pertaining to direct/indirect report messages for nodes within its four-hop neighborhood.

The algorithm operates as follows:

- Initially, the source does a local broadcast of the message.

- Each neighbor i of the source immediately commits to the the the first value v it heard from the source, and then locally broadcasts it once in a $COMMITTED(i, v)$ message.

- Hereafter, the following algorithm is followed by each node j (including those involved in the previous two steps):

On receipt of a $COMMITTED(i, v)$ message from neighbor i, record the message, and locally broadcast a $HEARD(j, i, v)$ message.

On receipt of a $HEARD(k, i, v)$ message from a neighbor k, record the message, and locally broadcast a $HEARD(j, k, i, v)$ message.

On receipt of a $HEARD(l, k, i, v)$ message, record the message, and locally broadcast a $HEARD(j, l, k, i, v)$ message.

On receipt of a $HEARD(g, l, k, i, v)$ message, record the message, but do not re-propagate.

On committing to a value v, do a one-time local broadcast of $COMMITTED(j, v)$.

A node j commits to a value v if it has not already comitted to a value, and it reliably determines that at least $t+1$ nodes lying in some single neighborhood have committed to v. j is said to have reliably determined the value committed to by node i if one of the following conditions holds:

- i is its neighbor, and j heard $COMMITTED(i, v)$ directly. In this case, there is no cause for doubt as to the value announced by node i, since no other node is capable of spoofing i's address, and collisions are ruled out.

- j heard indirect reports of i having committed to a particular value v through $t+1$ node-disjoint paths that all lie within *some single neighborhood*. The indirect reports are obtained through the *HEARD* messages that propagate via upto three intermediate nodes (i.e., upto four hops from the node that sent the *COMMITTED* message), and the path information is obtained from these messages (as each forwarding node affixes its identifier to the message).[4]

Theorem 20. *No non-faulty node shall commit to a wrong value by following the above algorithm.*

Proof. The proof is by contradiction. Consider the first non-faulty node, say j, that makes a wrong decision to commit to a value v. Evidently, j cannot be a neighbor of the source. This implies it reliably determined that $t+1$ already committed nodes lying in some single neighborhood N_1 had committed to v. Since reliable determination of a node i having committed to a value v involves hearing i directly or hearing indirect reports (that i committed to v) via at least $t+1$ node-disjoint paths lying in some single neighborhood N_2, and since the number of faults in N_2 may be at most t, all these paths cannot have relayed the wrong value, and v must indeed be the committed-to value announced by i. Thus, no non-faulty

---

[4]Note that a faulty intermediate node can affix a false identity for itself, or alter the affixed identities of previous nodes listed in the message it is forwarding (these are part of the message content, which can be altered). This does not cause a problem as the identity of the last faulty node (let us call it x) on the forwarding path will always be revealed to j (either x is j's neighbor, in which case j knows its identity as MAC addresses cannot be spoofed, or there is some other non-faulty node on the forwarding path after x which knows the message was relayed through x since it knows the correct MAC adddress of the previous hop node). Thus, even if x has affixed a wrong identity for itself in the message path information, the next non-faulty node can detect this and rectify the situation, and subsequent relays are all non-faulty. Therefore, j will know that x lies on the path. Hence, even if x has altered the identities of nodes before it on the forwarding path, this is acceptable, as j will not consider any other message with a path through x, and resultantly given only t faulty nodes in the neighborhood, at most t out of the $t+1$ paths can involve faulty information.

node can make a wrong determination of what value each of the $t+1$ nodes in N_1 committed to (or claimed to commit to). Since j is the first non-faulty node to make a wrong decision, the non-faulty nodes amongst the $t + 1$ nodes could not have made a wrong decision, and a value committed to and announced by such nodes must be correct. Also, all of the $t + 1$ nodes cannot be faulty, as no more than t nodes in any neighborhood may exhibit Byzantine failure. Therefore, at least one other non-faulty node previously committed to v. So, it must indeed be the correct value, else we would obtain a contradiction. □

Theorem 21. *Each non-faulty node is eventually able to commit to the correct value.*

Proof. We prove that each non-faulty node will be able to meet the conditions stipulated by the algorithm for committing to the correct value. The essence of the proof lies in showing that each node j other than the direct neighbors of $(0,0)$ is connected to at least $2t + 1$ nodes that lie in some single neighborhood N_1, such that the connectivity to each such node is through $2t + 1$ node-disjoint paths that all lie in some neighborhood N_2, and the nodes in N_1 are able to commit to the correct value before node j has done so.

The proof is by induction.

Base Case: All non-faulty nodes in $nbd(0,0)$ are able to commit to the correct value. This follows trivially from the assumed model, since they hear the origin directly.

Inductive Hypothesis: If all non-faulty neighbors of a node located at (a,b) i.e. all non-faulty nodes in $nbd(a,b)$ are able to commit to the correct value, then all non-faulty nodes in $pnbd(a,b)$ are able to commit to the correct value.

Proof of Inductive Hypothesis: We show that each node in $pnbd(a,b) \setminus nbd(a,b)$ is able to reliably determine the value committed to by $2t + 1$ nodes in $nbd(a,b)$. Since no more than t of these can be faulty, this guarantees that the node will become aware of $t+1$ nodes in $nbd(a,b)$ having committed to a (the correct) value, and will also commit to it. In order to show this, we prove that each node is connected to at least $2t+1$ nodes in $nbd(a,b)$ either directly, or through $2t + 1$ node disjoint paths that all lie entirely within some single neighborhood. Thus at least $t + 1$ of these paths are guaranteed to be fault-free and shall allow communication of the correct value.
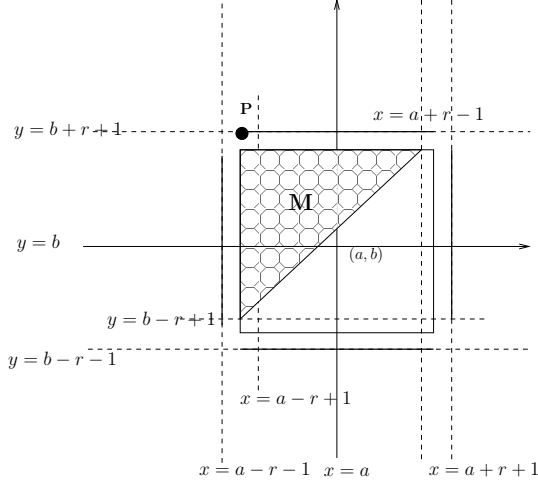
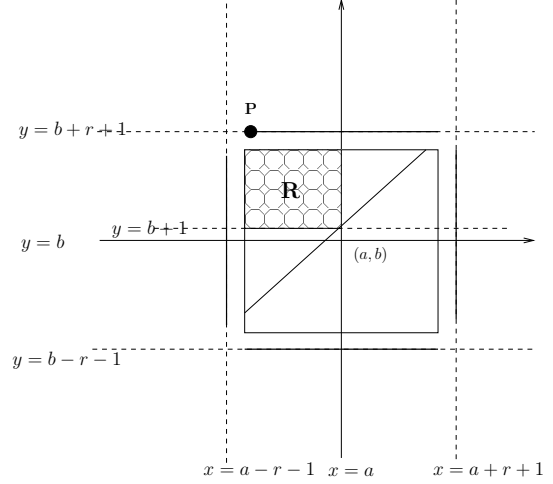Figure 8.8: Nodes in $nbd(a,b)$ whose committed values P can reliably determine

Figure 8.9: Nodes in $nbd(a,b)$ that are immediate neighbors of P

We show this for a corner node in $pnbd(a,b) \setminus nbd(a,b)$, i.e., the node marked P (which is located at $(a-r, b+r+1)$) in Fig. 8.8. This represents the worst case. For all other nodes in $pnbd(a,b) \setminus nbd(a,b)$, the condition can be seen to be achieved via a similar argument. We briefly discuss this later.

We show that node P is able to reliably determine the values committed to by the nodes in the shaded region M in Fig. 8.8. Region M comprises $\{(a-r+p, b-r+q) | 2r \geq q > p \geq 0\}$ and hence has $r(2r+1)$ nodes.

The first observation is that P can directly hear the nodes in the shaded sub-region R in Fig. 8.9, comprising $\{(x,y) | (a-r) \leq x \leq a; (b+1) \leq y \leq (b+r)\}$ (this constitutes $r(r+1)$ nodes), and so can trivially reliably determine the value they committed to. The remaining sub-regions are depicted in Fig. 8.10 as U (comprising $\frac{1}{2}r(r-1)$ nodes), S_1 (comprising r nodes), and S_2 (comprising $\frac{1}{2}r(r-1)$ nodes).

We now explicitly prove existence of suitable node-disjoint paths for nodes that lie in the upper triangular region U in Fig. 8.10. Any node N in this region may be considered located at $(a+p, b+q)$ (Fig. 8.11), such that $r \geq q > p \geq 1$ in this region. We show the existence of $r(2r+1)$ node-disjoint paths between N and P, that all lie within the same single neighborhood (centered at $(a, b+r+1)$), and indicated by the square with dark outline in Fig. 8.12. For greater clarity, the spatial extents of various demarcated regions used in the following argument are tabulated in Table 8.1.

Consider Fig. 8.12. The region marked A comprises $\{(x,y) | (a+p-r) \leq x \leq a; (b+1) \leq$
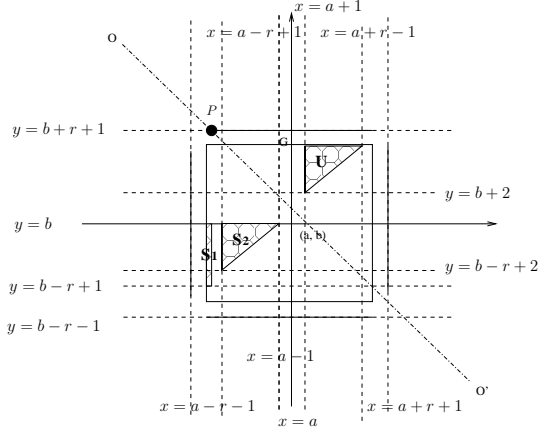
Figure 8.10: Nodes in $nbd(a,b)$ to which P has sufficient connectivity
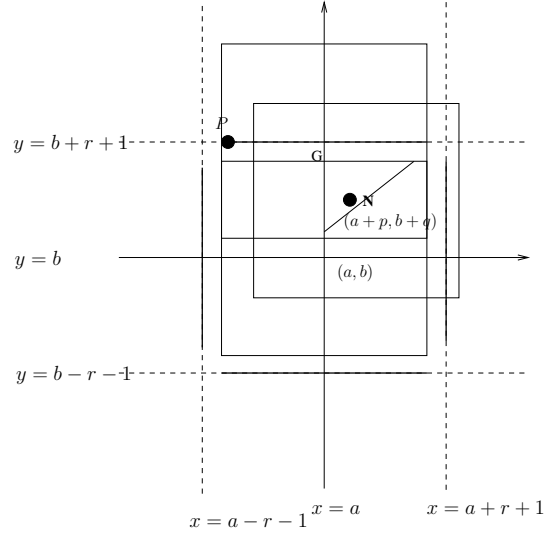
Figure 8.11: A node N in Region U

Region	x-extent	y-extent
A	$(a+p-r) \leq x \leq a$	$(b+1) \leq y \leq (b+q+r)$
B_1	$(a+1) \leq x \leq (a+p-1)$	$(b+1) \leq y \leq (b+q+r)$
B_2	$(a+1-r) \leq x \leq (a+p-1-r)$	$(b+1) \leq y \leq (b+q+r)$
C_1	$(a+p+1) \leq x \leq (a+r)$	$(b+q+1) \leq y \leq (b+r+1)$
C_2	$(a+p+1-r) \leq x \leq a$	$(b+q+1+r) \leq y \leq (b+1+2r)$
D_1	$(a+p) \leq x \leq (a+p+r-q)$	$(b+r+q-p+1) \leq y \leq (b+r+q)$
D_2	$(a+1) \leq x \leq (a+p)$	$(b+1+r+q) \leq y \leq (b+1+2r)$
D_3	$(a+1-r) \leq x \leq (a+p-r)$	$(b+1+r+q) \leq y \leq (b+1+2r)$
J	$(a-2r) \leq x \leq a$	$(b+1) \leq y \leq (b-p+r)$
K_1	$(a-2r) \leq x \leq a$	$(b-p+1) \leq y \leq b$
K_2	$(a-2r) \leq x \leq a$	$(b-p+r+1) \leq y \leq (b+r)$

Table 8.1: Spatial Extents of Various Regions
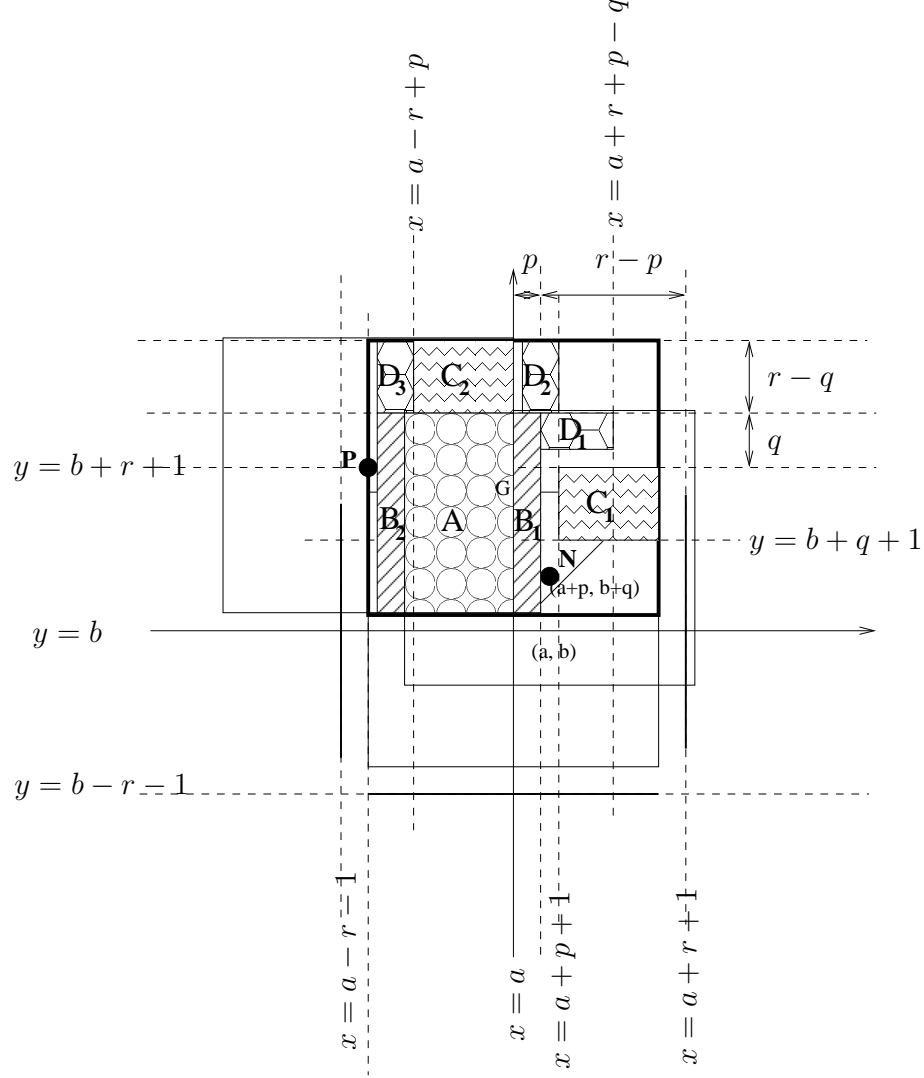
Figure 8.12: Construction depicting node-disjoint paths between N and P

$y \leq (b + q + r)\}$, and nodes in this region are neighbors of both N and P. Thus, there are $(r - p + 1)(r + q)$ paths of the form $N \rightarrow A \rightarrow P$ that comprise one intermediate node each.

The region B_1 comprises $\{(x, y) | (a + 1) \leq x \leq (a + p - 1); (b + 1) \leq y \leq (b + q + r)\}$, and falls in $nbd(N)$ (recall that N is located at $(a + p, b + q)$). The region B_2 comprises $\{(x, y) | (a + 1 - r) \leq x \leq (a + p - 1 - r); (b + 1) \leq y \leq (b + q + r)\}$, and falls in $nbd(P)$. As may be seen, B_2 is obtained by a translation of B_1 to the left by r units. Thus, there is a one-to-one correpondence between a node at (x, y) in B_1 and a node at $(x - r, y)$ in B_2, such that the nodes in each pair are neighbors. This yields $(p - 1)(r + q)$ paths of the form
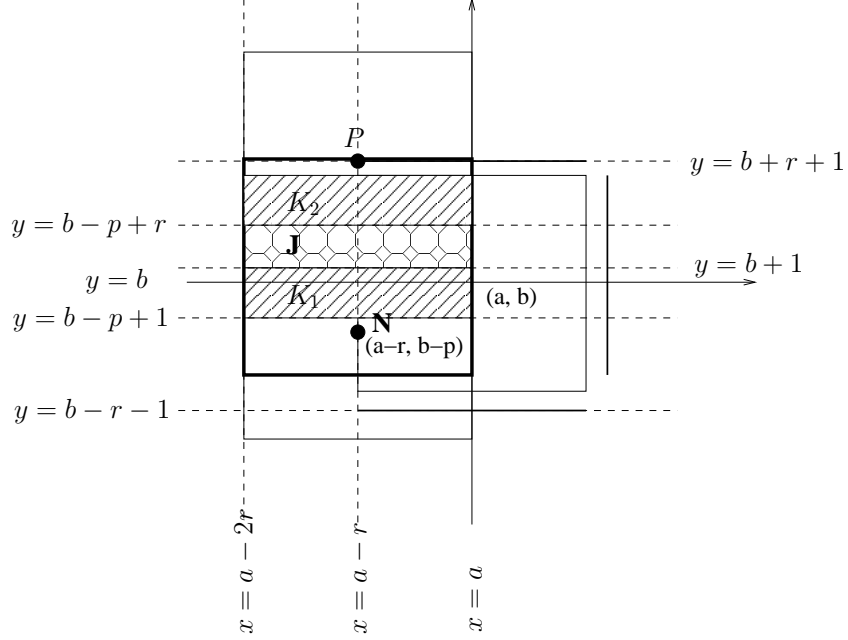
Figure 8.13: Connectivity between P and nodes in S_1

$N \to B_1 \to B_2 \to P$.

Region C_1 comprises $\{(x,y)|(a+p+1) \le x \le (a+r); (b+q+1) \le y \le (b+r+1)\}$ and thus falls within $nbd(N)$. Region C_2 comprises $\{(x,y)|(a+p+1-r) \le x \le a; (b+q+1+r) \le y \le (b+1+2r)\}$ and falls within $nbd(P)$. It may be seen that there is a one-to-one correspondence between any node at (x,y) in C_1 and the node at $(x-r, y+r)$ in C_2, with the paired nodes being neighbors. Hence there exist $(r-p)(r-q+1)$ paths of the form $N \to C_1 \to C_2 \to P$ that comprise two intermediate nodes each.

Region D_1 comprises $\{(x,y)|(a+p) \le x \le (a+p+r-q), (b+r+q-p+1) \le y \le (b+r+q)\}$, and falls in $nbd(N)$. Region D_2 comprises $\{(x,y)|(a+1) \le x \le (a+p); (b+1+r+q) \le y \le (b+1+2r)\}$. Region D_3 comprises $\{(x,y)|(a+1-r) \le x \le (a+p-r); (b+1+r+q) \le y \le (b+1+2r)\}$, and falls in $nbd(P)$. We note that regions D_1, D_2 and D_3 have exactly the same number of nodes each. Besides, the regions D_1 and D_2 are mutually located in a manner that each node in D_2 is a neighbor of each node in D_1 (maximum distance between any two nodes $\le r$). Hence, any one-to-one pairing of nodes in D_1 with nodes in D_2 is valid. Further, a node located at (x,y) in D_2 has a one-to-one correpondence with a node $(x-r, y)$ in D_3. Hence, there are $p(r-q+1)$ paths of the form $N \to D_1 \to D_2 \to D_3 \to P$ that comprise three intermediate nodes each (Fig. 8.12). Thus the $r(2r+1)$ node-disjoint

201

paths are obtained.

We now consider nodes in regions S_1 and S_2 depicted in Fig. 8.10.

$S_1 = \{(a - r, b - p)|0 \leq p \leq (r - 1)\}$. It can be shown that P has $r(2r + 1)$ disjoint paths to each node in S_1, as depicted in Fig. 8.13. Any node N in S_1 is located at $(a - r, b - p)$ where $0 \leq p \leq (r - 1)$. Consider region J comprising $\{(x, y)|(a - 2r) \leq x \leq a; (b + 1) \leq y \leq (b - p + r)\}$. All nodes in J are common neighbors of N and P, and provide $(r - p)(2r + 1)$ paths of the form $N \rightarrow J \rightarrow P$. Region K_1 comprises $\{(x, y)|(a - 2r) \leq x \leq a; (b - p + 1) \leq y \leq b\}$, and falls enirely within $nbd(N)$. Region K_2 is $\{(x, y)|(a - 2r) \leq x \leq a; (b - p + r + 1) \leq y \leq (b + r)\}$, and falls in $nbd(P)$. For each node (x, y) falling in K_1, there is a one-to-one correspondence with a node $(x, y + r)$ in K_2, and thus we obtain $p(2r + 1)$ paths of the form $N \rightarrow K_1 \rightarrow K_2 \rightarrow P$. This yields a total of $r(2r + 1)$ paths (all lying entirely within $nbd(a - r, b + 1)$), as depicted in Fig. 8.13.

Region S_2 comprises $\{(a - q, b - p)|(r - 1) \geq q > p \geq 0\}$. For the nodes in S_2, observe that each node $(a - q + 1, b - p + 1)$ in S_2 possesses the same relative position w.r.t. P as the node $(a + p, b + q)$ in region U of Fig. 8.10 (note the axial symmetry about axis OO'), and due to the symmetric structure of the network, shall enjoy exactly the same connectivity properties to P as the node $(a + p, b + q)$ in region U. Since we have already shown existence of sufficient connectivity for those nodes, the same holds for nodes in S_2.

The inductive hypothesis, along with the base case, suffices to show that every non-faulty node will eventually commit to the correct message value, since starting at $(0, 0)$, one can cover the entire infinite grid by moving up, down, left and right. Thus, non-faulty nodes in the neighborhood of every grid point can be shown to be able to eventually determine the broadcast value.

$\square$

Non-worst Case Location of P We briefly discuss how the connectivity argument holds for all $P \in pnbd(a, b) \setminus nbd(a, b)$. We consider non-worst case locations of $P \in \{(a - r + l, b + r + 1)|1 \leq l \leq r\}$. For all other locations, the argument holds by symmetry. The situation is depicted in Fig. 8.14. One may consider P to be translated to the right by l units from its worst case location at $(a - r, b + r + 1)$. Then, region R that lies in direct range of P (recall from Fig. 8.9) now comprises $r(r + l + 1)$ nodes. If we also translate regions U, S_1, and S_2 by l units each to the right, they preserve their relative positions and
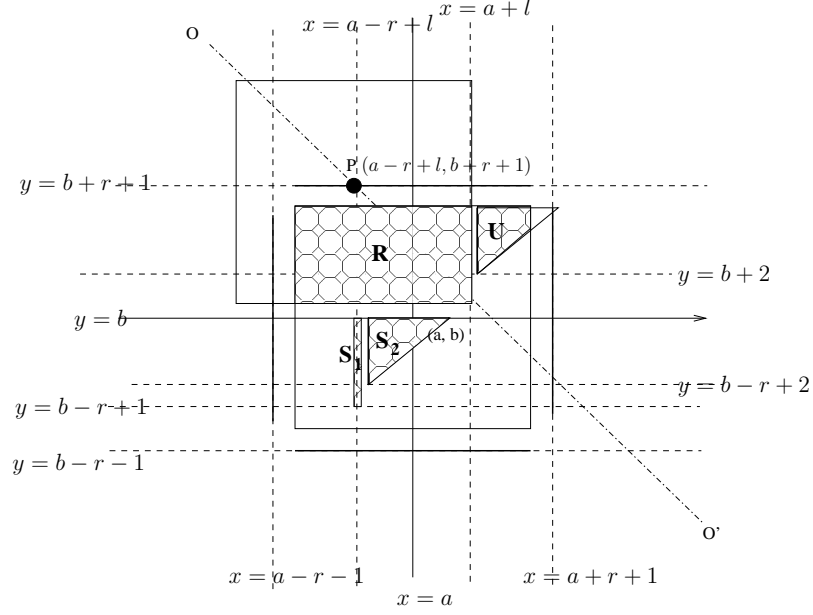
Figure 8.14: Non-worst Case Location of P

hence connectivity to P. However, now $\frac{1}{2}l(l-1)$ nodes from U fall out of $nbd(a,b)$, but this is more than compensated by the increase of rl nodes in region R. Thus, if we count the number of nodes in $nbd(a,b) \cap U$, $nbd(a,b) \cap S_1$, and $nbd(a,b) \cap S_2$, it can be shown that they are at least $r(r-l)$ in number. Together with the $r(r+l+1)$ nodes in region R, they provide at least $r(2r+1)$ nodes to which P is connected either directly or via $2t+1$ node-disjoint paths all lying within some single neighborhood.

8.7.1 Comparison of the Two Algorithms

The algorithm described in this section is based on the stronger condition that every node in $pnd(a,b) \setminus nbd(a,b)$ has $2t+1$ node-disjoint paths, all lying within some single neighborhood, to each of $2t+1$ nodes in $nbd(a,b)$. The algorithm described in Section 8.4 relies on a simpler condition, and yet suffices to ensure reliable broadcast. It is also more efficient in terms of greater localization of propagated messages. The alternative algorithm is still of interest, as the particular localized connectivity property may possibly find use in distributed operations other than reliable broadcast.

203

8.8 Discussion

In this chapter, we stated and proved results regarding the number of Byzantine and crash-stop failures that may be tolerated in an idealized wireless network without rendering reliable broadcast impossible. We considered a locally-bounded adversarial model where the adversary is free to choose faulty nodes, so long as the placement satisfies the constraint that no neighborhood has more than t faults. However, in the presence of channel errors etc., the *reliable local broadcast* assumption that underlies these results is not trivial to realize. Thus, implementation of a reliable broadcast service based on this model would require efficient implementation of a *reliable local broadcast primitive* that operates under realistic network conditions. In Chapter 10, we consider this issue in some detail.

8.9 Future Directions

In this chapter, we described results for achievability of reliable broadcast with locally bounded failures. However, we did not study the efficiency of the algorithms. Thus, it would be of interest to determine the optimal communication complexity for achieving reliable broadcast for the grid network, as well as a wider class of network models. Moreover, our focus was on a single broadcast instance; in typical application scenarios, the broadcast operation will occur many times. In such scenarios, incorporating fault-detection mechanims can allow one to achieve weaker properties similar to the self-adjusting Byzantine agreement of [129], which are often sufficient to meet reliability requirements. This is a particularly promising approach in the wireless context, since the broadcast nature of the wireless medium may make it easier to detect faulty behavior. Therefore, it is very relevant to consider designing such algorithms for wireless network scenarios.

Chapter 9

Reliable Broadcast with Probabilistic Failures

In this chapter, we consider the problem of reliable broadcast in wireless networks with probabilistic failures. Our primary focus is on Byzantine failures, but we have also briefly addressed the case of crash-stop failures. We begin by introducing the model and notation in Section 9.1, and then summarize the chapter results in Section 9.2. We describe a general necessary condition in Section 9.3. We present necessary and sufficient conditions for reliable broadcast in a toroidal grid network in Section 9.4 and Section 9.5 respectively, assuming the L_∞ distance metric. A sufficient condition for random networks is presented in Section 9.6. Results for grid networks with crash-stop failures are discussed in Section 9.7. In Section 9.8 we discuss how the L_∞ metric results can be used to obtained results for the L_2 metric, and in Section 9.9, we argue for the validity of the results even in non-toroidal networks. We also identify an interesting but intuitive similarity in the structure of the results (previously known results, as well as the results presented in this chapter) for a set of related problems pertaining to connectivity and reliable broadcast. This is discussed in Section 9.10.

9.1 Preliminaries

We consider two spatial layout models for the network:

1. A regular grid layout, where nodes are located on a two-dimensional square grid (each grid unit is a 1×1 square). We shall refer to this as a *grid network*.

2. A network in which the node locations are independently and identically (i.i.d.) distributed over the deployment region. We shall refer to this as a *random network*.

In both models, the network is assumed to be deployed over a $\sqrt{n} \times \sqrt{n}$ square region. Each node is assumed to be aware of the locations of all nodes within its transmission range.

Recall the definition of the reliable broadcast problem with a designated source in Chapter 7. For the results in this chapter, we assume that any node in the entire network can be the designated source and can originate a broadcast message. Given such a broadcast instance, if even one non-faulty node (in either model) fails to make a valid value determination, the broadcast is deemed to have failed. Thus, reliable broadcast is said to fail in a given fault configuration if it fails for at least one possible choice of the designated broadcast source.

For a given broadcast instance, once an origin/source is designated, it is identified as $(0,0)$. All nodes can then be uniquely identified by their coordinate location (x, y) w.r.t. this origin. In the grid network model, the node coordinates are always *integers*, while for random networks they are *real* numbers. All nodes have a common transmission radius $r(n,p)$ (often abbreviated as r). For grid networks, we assume that $r(n,p)$ is an integer, and for random networks it is allowed to be any real number.

In the toroidal grid network, each node has the same number of neighbors (i.e., the same degree). We use $d(n,p)$ (often abbreviated as d) to denote the common node degree for this model. The neighbor-set of a node u, *including itself*, is denoted by $nbd(u)$. The set of neighbors excluding itself is denoted by $nbd'(u) = nbd(u) \setminus \{u\}$.

For the grid network, in the L_∞ metric, the degree of a node is $4r^2 + 4r$, while the population of a neighborhood (including the neighborhood center) is $d + 1 = 4r^2 + 4r + 1$. Thus, the minimum node degree is $d_{min} = 8$, corresponding to $r = 1$.

For succint description, we also define a term $pnbd(x, y)$ where $pnbd(x, y) = nbd(x - 1, y) \cup nbd(x + 1, y) \cup nbd(x, y - 1) \cup nbd(x, y + 1)$. Intuitively $pnbd(x, y)$ denotes the *perturbed neighborhood* of (x, y), obtained by perturbing the center of the neighborhood by ± 1 along the x and y axes. We use $Bernoulli(p)$ to denote a Bernoulli random variable with parameter p.

A random failure mode is assumed wherein each node can fail with probability p independently of other nodes. Failures are permanent. We primarily focus on Byzantine failures. In the Byzantine failure mode, a faulty node can behave arbitrarily, in contrast to crash-stop failures, where a faulty node simply stops functioning. As stated in Chapter 7, we assume that the Byzantine nodes cannot spoof addresses or cause collisions, i.e., the MAC layer is assumed fault-free, and the Byzantine faults reside only in higher layers of the

protocol stack.[1] Note that while the *occurrence* of the permanent failures is probabilistic, the failed Byzantine nodes can thereafter choose to behave in a worst-case manner (i.e., collude and modulate the messages they send to cause most confusion to non-faulty nodes). The non-faulty nodes do not know which nodes have failed. The wireless channel conforms to the reliable local broadcast assumption described in Chapter 7.

When we use the term *critical* transmission range for reliable broadcast, we imply the smallest transmission range that can ensure reliable broadcast with high probability (w.h.p.).

Thus:

- When we say that the critical transmission range is $\Omega(f(n,p))$, we imply that:

  $$\exists\, c_1 > 0, \text{ such that when } r(n,p) \leq c_1 f(n,p) : \lim_{n \to \infty} \Pr[\text{reliable broadcast achievable}] < 1$$

  Thus, the transmission range must necessarily be greater than $c_1 f(n,p)$ for reliable broadcast to be achievable w.h.p.

- When we say the critical transmission range is $O(f(n,p))$, we imply that:

  $$\exists\, c_2 > 0, \text{ such that when } r(n,p) \geq c_2 f(n,p) : \lim_{n \to \infty} \Pr[\text{reliable broadcast achievable}] = 1$$

  Thus, the smallest transmission range needed to achieve reliable broadcast is no more than $c_2 f(n,p)$.

- When we say that the critical range is $\Theta(f(n,p))$, we imply that it is $\Omega(f(n,p))$ and $O(f(n,p))$.

In a grid network, with the L_∞ metric (discussed in Section 9.1), the node degree is exactly determined by specifying the transmission range. Hence, we can define the notion of *critical* degree correponding to the critical transmission range. Thus:

- When we say that the critical degree is $\Omega(g(n,p))$, we imply that:

  $$\exists\, a_1 > 0, \text{ such that when } d(n,p) \leq a_1 g(n,p) : \lim_{n \to \infty} \Pr[\text{reliable broadcast achievable}] < 1$$

---

[1] A methodology to handle a bounded number of collisions and address-spoofing was proposed in [58] for a locally bounded fault model. It might be possible to adapt it to handle the random failure model. This requires further investigation.

This yields a *necessary* condition.

- When we say that the critical degree is $O(g(n,p))$, we imply that:

  $$\exists \ a_2 > 0, \text{ such that when } d \geq a_2 g(n,p) : \lim_{n \to \infty} \Pr[\text{reliable broadcast achievable}] = 1$$

  This yields a *sufficient* condition.

- When we say that the critical degree is $\Theta(g(n,p))$, we imply that it is $\Omega(g(n,p))$ and $O(g(n,p))$

In a random network, the degrees of individual nodes can vary; however, it is possible to define a notion of *critical* average degree, which is the average degree corresponding to the critical transmission range.

9.2 Summary of Results

In this chapter, we show that:

1. In a network of n nodes deployed in a regular grid pattern, when nodes exhibit Byzantine failure with failure probability $p < \frac{1}{2}$ (see later sections for precise range of validity), the *critical* node degree (defined in Section 9.1) for asymptotic achievability of reliable broadcast is $\Theta\left(d_{min} + \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}}\right)$. This may alternatively be stated as $\Theta\left(d_{min} + \frac{\ln n}{D(Q_{\frac{1}{2}} || P)}\right)$ where $Q_{\frac{1}{2}}$ denotes the *Bernoulli*($\frac{1}{2}$) distribution, P denotes the *Bernoulli*(p) distribution, and $D(Q||P)$ denotes the *relative entropy* (or Kullback-Leibler distance) between distributions Q and P.

2. In a network of n nodes located uniformly at random over the network region, when nodes exhibit Byzantine failure with failure probability $p < \frac{1}{2}$, the critical average node degree for reliable broadcast is $O(\ln n + \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}})$(also expressible as $O\left(\frac{\ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}}\right)$ for this regime).

3. For crash-stop failures in a grid deployment, the problem of reliable broadcast is equivalent to connectivity in the presence of faults. For this case, we have derived
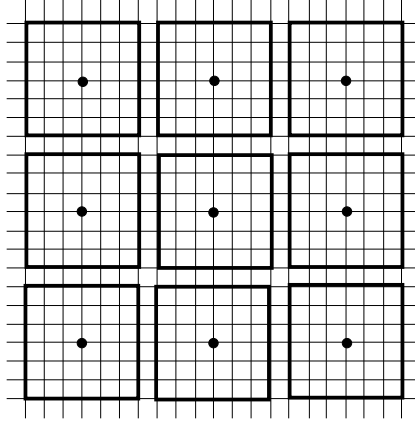
Figure 9.1: Division of network into disjoint neighborhoods

results showing that the critical node degree is $\Theta\left(d_{min} + \frac{\ln n}{\ln \frac{1}{p}}\right)$ with failure probability $p < 1$ (see later sections for precise range of validity). Our results improve upon previous results for crash-stop failures in a grid proved in [100] in the regime $p \to 0$.

A preliminary version of the chapter results was reported in [9].

9.3 General Necessary Condition for Byzantine Failures

In this section, we show that if at least half the neighbors of a non-faulty node not in $nbd(s)$ are faulty in the Byzantine sense, then the faulty nodes can make it commit to the wrong broadcast value with probability at least $\frac{1}{2}$. We remark that it is possible for a node to refrain from committing to any value (in which case it would not commit to the wrong value). However, if a non-faulty node does not commit to any value, then this implies failure of the reliable broadcast operation, and from the perspective of *achievability* of broadcast this is no better than committing to a wrong value. Thus, we focus on the case where a node does indeed commit to some value.

Theorem 22. *Under the assumption that all message values are equally likely, if a non-faulty node $u \notin nbd(s)$ has at least half faulty neighbors, then it can be made to commit to an erroneous value with probability at least $\frac{1}{2}$.*

Proof. Assume that the message is drawn from $\{0, 1\}$. A non-faulty node u which is not an immediate neighbor of the source must rely on messages received from its neighbors. Recall that $nbd'(u) = nbd(u) \setminus \{u\}$ and $d = |nbd'(u)|$.

First consider any deterministic function that takes as argument messages received from all neighbors and outputs one of 0 or 1. Corresponding to each fault configuration C_1 with $t \geq \frac{d}{2}$ faults in $nbd'(u)$ (this also implies t faults in $nbd(u)$ as u is non-faulty), there is another configuration C_2 with t faults in $nbd'(u)$, such that all non-faulty nodes in C_1 are faulty in C_2, while the non-faulty nodes in C_2 were all faulty in C_1. Then the faulty nodes can modulate their message-sending behavior so that u is unable to distinguish between the case where the correct broadcast value was 0 and fault configuration was C_1 and the case when the correct value was 1 and the fault configuration was C_2 (recall that once failure has happened, the faulty nodes can exhibit worst-case behavior).

Stated formally: suppose $\mathcal{S}_1 \subseteq nbd'(u)$ is the set of faulty neighbors in C_1, and $\mathcal{S}_1^c = nbd'(u) \setminus \mathcal{S}_1$ is its complement, i.e., the set of non-faulty neighbors. Then we know that $|\mathcal{S}_1| \geq \lceil \frac{|nbd'(u)|}{2} \rceil \geq |\mathcal{S}_1^c|$.

Consider a fault configuration C_2 in which the set of faulty neighbors is $\mathcal{S}_2 = \mathcal{S}_1^c \cup \mathcal{V}$ where $\mathcal{V} \subseteq \mathcal{S}_1$ is some subset of $\mathcal{S}_1$ that satisfies $|\mathcal{V}| = |\mathcal{S}_1| - |\mathcal{S}_1^c|$. Let $\mathcal{S}_2^c$ denote the complement of $\mathcal{S}_2$. It is easy to see that $|\mathcal{S}_1| = |\mathcal{S}_2|$. Consider the case where the correct value is 0, and fault configuration is C_1. Then all nodes in $\mathcal{S}_1$ can behave as though the value were 1, while the nodes in $\mathcal{S}_1^c$ will always act according to value 0. Now suppose the correct value is 1, and the fault configuration is $\mathcal{C}_2$. Then the faulty nodes in $\mathcal{S}_1^c \subseteq \mathcal{S}_2$ behave as though the value were 0, while nodes in $\mathcal{V} = \mathcal{S}_2 \setminus \mathcal{S}_1^c$ act as per the correct value 1. The non-faulty nodes in $\mathcal{S}_2^c$ always act as per value 1. From the viewpoint of node u, the two situations are indistinguishable.

Next consider the possibility of using a probabilistic decision rule. Given a set of messages received from neighbors, we need to consider the conditional probability that the value is 0 or 1. From the above discussion it is clear that for a given set of received messages from neighbors, there exists a pair of fault configurations, and associated faulty-node behavior, with the same number of faulty neighbors, where the correct message values are different. Since failures are i.i.d. with probability p, and each value 0 or 1 is equiprobable, u cannot hope to choose the correct one with a probability greater than half.

It is not hard to see that if the message can have more than two possible (equiprobable) values, it cannot increase the probability of correct choice.

$\square$

If the failure probability p is at least $\frac{1}{2}$, it can be seen that the probability that at least half the neighbors of a given node are faulty is at least $\frac{1}{2}$ (for even node degree, this follows from Lemma 50; for odd degree, we can first argue for $p = \frac{1}{2}$ and then use a monotonicity argument). Therefore, it is only relevant to study the achievability of broadcast for $p < \frac{1}{2}$.

9.4 Byzantine Failures in a Grid Network: Necessary Condition

Note that when $p = 0$, it is still necessary to ensure that each node has non-zero degree for broadcast to be possible, and this requires that r be set to 1 (and hence $d = d_{min} = 8$). Thus, when $p = 0$, it trivially follows that the node degree must be at least $\max\{d_{min}, \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}}\}$ (we adopt the standard convention that $x \log \frac{x}{0} = \infty$ for any $x > 0$; we also adopt the convention that $\frac{y}{\infty} = 0$ for any finite $y > 0$).

Hence, the case of interest is when $p > 0$. It is easy to see that $r \geq 1$ (correspondingly $d \geq d_{min} = 8$) is necessary for any p.

Theorem 23. *Assuming the L_∞ distance metric, in a grid network where nodes can fail (in a Byzantine sense) independently with probability p such that $0 < p \leq \frac{1}{2} - \frac{1}{\ln n}$, if the node degree is $d \leq \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}}$:*

$$\Pr[\ \text{reliable broadcast fails}\ ] = 1$$

Proof. It is evident that $r(n, p)$ must be at least 1 for reliable broadcast, else all nodes in the grid are isolated. Thus $d(n, p)$ must be at least $d_{min} = 8$. Therefore, in the rest of the proof, we only need to consider the case where $\frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}} \geq d_{min}$, and $r(n, p)$ is set to at least 1.

Any failure probability $p \leq \frac{1}{2} - \frac{1}{\ln n}$ can be expressed as $p = \frac{1}{2} - y$ for suitable $\frac{1}{\ln n} \leq y < \frac{1}{2}$.

It can be seen that:

$$\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)} = \ln \frac{1}{1 - 2y} + \ln \frac{1}{1 + 2y}$$
$$= \ln \frac{1}{1 - 4y^2} \geq 4y^2 \quad \text{(noting that } 4y^2 < 1 \text{ and applying Fact 1)} \quad (9.1)$$
$$\geq \frac{4}{(\ln n)^2}$$

211

Resultantly:

$$d \leq \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}} \leq \frac{\ln n}{\frac{4}{(\ln n)^2}} = \frac{(\ln n)^3}{4} < (\ln n)^3 \tag{9.2}$$

Furthermore, it is evident that:

$$\frac{\ln n}{2} + 6 \ln \ln n \leq \ln n - 4 \ln \ln n \quad \text{for sufficiently large } n \tag{9.3}$$

Consider a particular node j in the network. From Theorem 22, it follows that if j is non-faulty, but more than half of its neighbors are faulty, reliable broadcast will fail with probability at least $\frac{1}{2}$.

We know that there are d neighbors of j, and each may fail independently with probability p. Let $I_{jk}(1 \leq k \leq d)$ denote an indicator variable corresponding to neighbor k of j (enumerated in some order), such that $I_{jk} = 1$ if k is faulty, and 0 otherwise. Then $Y_j = \sum_{k \in nbd'(j)} I_{jk}$ denotes the number of failed neighbors of j. Y takes values from $0, 1, ..., d$, and $E[Y] = pd$. Note that in the L_∞ metric, d is always even, and $d \geq 8$ for all $r(n, p) \geq 1$. Also:

$$\Pr[Y_j \geq \frac{d}{2}] = \sum_{i=\frac{d}{2}}^{d} \binom{d}{i} p^i (1-p)^{(d-i)}$$

Let us simply consider the event $Y_j = \frac{d}{2}$. Then we can apply the lower bound from Lemma 56 as follows: the variables $I_{jk}(1 \leq k \leq d)$ are drawn from $\chi = \{0, 1\}$ as per distribution $P = Bernoulli(p)$, and the distribution corresponding to $Y_j = \frac{d}{2}$ is $Bernoulli(\frac{1}{2})$ (we shall refer to this as $Q_{\frac{1}{2}}$). $|\chi| = 2$, and $\frac{1}{(d+1)^{|\chi|}} = \frac{1}{(d+1)^2} > \frac{1}{\frac{3}{2}d^2} = \frac{2}{3}e^{-2\ln d}$. Thus, we obtain:

$$\Pr[Y_j \geq \frac{d}{2}] \geq \Pr[Y_j = \frac{d}{2}] \geq \frac{1}{(d+1)^{|\chi|}} e^{-d(D(Q_{\frac{1}{2}}||P))}$$

$$= \frac{1}{(d+1)^2} e^{-d(D(Q_{\frac{1}{2}}||P))} = \frac{1}{d^2(1+\frac{1}{d})^2} e^{-d(D(Q_{\frac{1}{2}}||P))}$$

$$> \frac{2}{3} e^{-d(D(Q_{\frac{1}{2}}||P)) - 2\ln d} \tag{9.4}$$

$$> \frac{2}{3} e^{-(\frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}})(\frac{1}{2}\ln \frac{1}{2p} + \frac{1}{2}\ln \frac{1}{2(1-p)}) - 6\ln \ln n} \quad \text{using (9.2)}$$

$$= \frac{2}{3} e^{-\frac{1}{2}\ln n - 6\ln \ln n} \geq \frac{2(\ln n)^4}{3n} \quad \text{using (9.3)}$$

Let us denote the L.H.S. of the above equation, i.e., $\Pr[Y_j \geq \frac{d}{2}]$, by q.

$$\Pr[\,j \text{ non-faulty; at least half } nbd(j) \text{ faulty }] \geq (1-p)q > \frac{1}{2}\left(\frac{2(\ln n)^4}{3n}\right) = \frac{(\ln n)^4}{3n} \quad (9.5)$$

We mark out a subset of nodes j such that the neighborhoods of these nodes are all disjoint, as in Fig. 9.1. From Fact 3, the number of such nodes that we may obtain is $k \geq \frac{n}{2d}$ for large n (from (9.2), $d = o(n)$). In fact, it is not hard to see from the argument used in the statement of Fact 3 that the number of such nodes would exceed $\frac{n}{2d} + 1$ for large enough n. We can designate one such node as the broadcast source, and examine the probability that any of the remaining nodes ($k \geq \frac{n}{2d}$ in number) can be made to commit to the wrong broadcast value.

Let I_j be an indicator variable that takes value 1 if a node j is non-faulty and has at least half faulty neighbors. From (9.5), we know that $\Pr[I_j = 1] \geq \frac{(\ln n)^4}{3n}$. Furthermore, all the I_j's are independent.

Let I'_j be an indicator variable that takes value 1 if j is non-faulty but commits to a wrong value. From Theorem 22, we know that if a non-faulty node has half or more faulty neighbors, it can be made to commit to the wrong value with probability at least $\frac{1}{2}$. Thus $\Pr[I'_j = 1] \geq \frac{1}{2}\Pr[I_j = 1] \geq \frac{(\ln n)^4}{6n}$.

Let X be a random variable indicating the number of non-faulty nodes with half or more faulty neighbors that commit to the wrong value. Then $X = \sum I'_j$, and $E[X] = \sum \Pr[I'_j = 1] \geq \frac{(\ln n)^4}{6n}\left(\frac{n}{2d}\right) = \frac{(\ln n)^4}{12d} > \frac{\ln n}{12}$ ($\because d < (\ln n)^3$ from (9.2)). Therefore, we can choose a suitable constant $0 < \beta < 1$ (e.g., $\beta = \frac{1}{2}$) and apply the Chernoff bound in Lemma 53 to obtain:

$$\Pr[X > (1-\beta)E[X]] \geq 1 - e^{-\frac{\beta^2 E[X]}{2}}$$

$$\therefore \lim_{n \to \infty} \Pr[X > (1-\beta)E[X]] = 1 \qquad \because \lim_{n \to \infty} E[X] = \infty \tag{9.6}$$

This yields:

$$\lim_{n \to \infty} \Pr[\text{ reliable broadcast fails }] = 1$$

$\square$

In light of the prior observation about the necessity of $r(n,p)$ being at least 1 (i.e., $d(n,p)$ being at least d_{min}), and the result of Theorem 23, it follows that for all $p \leq \frac{1}{2} - \frac{1}{\ln n}$, if the node degree is less than $\max\{d_{min}, \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}}\}$, reliable broadcast fails w.h.p.
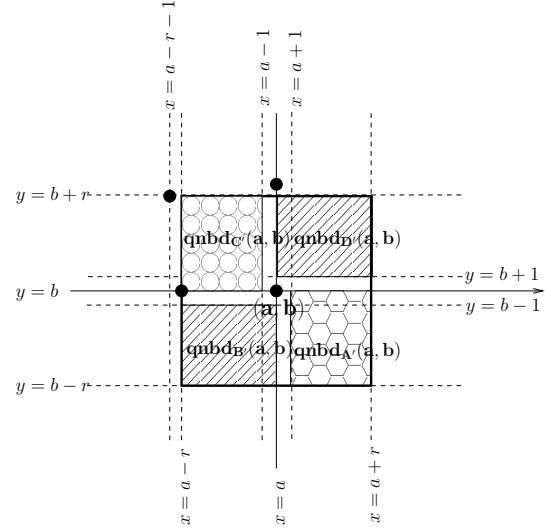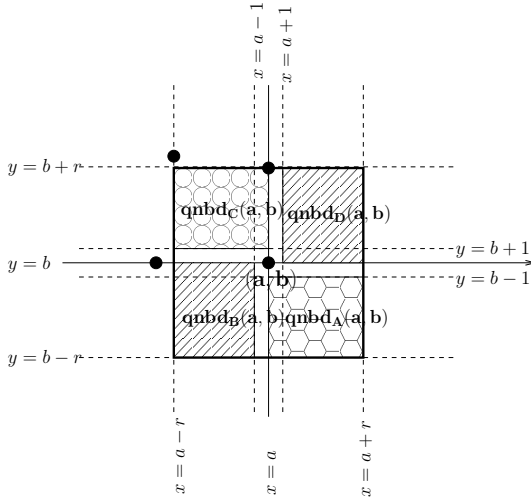
Figure 9.2: Depiction of $qnbd_A$, $qnbd_B$, $qnbd_C$, $qnbd_D$

Figure 9.3: Depiction of $qnbd_{A'}$, $qnbd_{B'}$, $qnbd_{C'}$, $qnbd_{D'}$

9.5 Byzantine Failures in a Grid Network: Sufficient Condition

We now state and prove a sufficient condition for the achievability of reliable broadcast in a grid network. Intuitively, the approach involves showing that if the degree of a node is sufficiently large, then the node can look at messages received from a constant fraction of its neighbors, and act upon the majority opinion in this subset; doing so will enable it to correctly determine the broadcast value, since a majority of the nodes in that subset will be non-faulty w.h.p.

Theorem 24. *Assuming L_∞ distance metric, in a grid network with Byzantine failure probability $p < \frac{1}{2}$, when $r(n,p)$ is chosen such that $d(n,p) = 4r^2 + 4r \geq \max\{d_{min}, 16\frac{\ln n}{\ln \frac{1}{p} + \ln \frac{1}{2(1-p)}}\}$ $= \max\{d_{min}, 8\frac{\ln n}{D(Q_{\frac{1}{2}}||P)}\}$:*

$$\lim_{n \to \infty} \Pr[\text{ reliable broadcast is achievable }] = 1$$

Note that when $\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)} \leq \frac{16 \ln n}{n}$, the degree expression exceeds the total network size n, the sufficient condition ceases to be relevant (as node degree $d(n,p)$ cannot exceed n). Note that such a value of $d(n,p)$ corresponds to a transmission range $r(n,p)$ of a node spans the entire network, effectively implying that the network is single-hop; due to the local

Region	x-extent	y-extent
$qnbd_A(a,b)$	$a \leq x \leq (a+r)$	$(b-r) \leq y \leq (b-1)$
$qnbd_B(a,b)$	$(a-r) \leq x \leq (a-1)$	$(b-r) \leq y \leq b$
$qnbd_C(a,b)$	$(a-r) \leq x \leq a$	$(b+1) \leq y \leq (b+r)$
$qnbd_D(a,b)$	$(a+1) \leq x \leq (a+r)$	$b \leq y \leq (b+r)$
$qnbd_{A'}(a,b)$	$(a+1) \leq x \leq (a+r)$	$(b-r) \leq y \leq b$
$qnbd_{B'}(a,b)$	$(a-r) \leq x \leq a$	$(b-r) \leq y \leq (b-1)$
$qnbd_{C'}(a,b)$	$(a-r) \leq x \leq (a-1)$	$b \leq y \leq (b+r)$
$qnbd_{D'}(a,b)$	$a \leq x \leq (a+r)$	$(b+1) \leq y \leq (b+r)$

Table 9.1: Spatial Extents of Quarter Neighborhoods

broadcast assumption, reliable broadcast is trivially achievable in a single-hop network.

Therefore, the sufficient condition is relevant only so long as $\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)} > \frac{16 \ln n}{n}$, and this is the case that we consider.

Case 1: $p = o(\frac{1}{n})$ By application of the union bound, the probability that at least one node fails is at most np. Since $p = o(\frac{1}{n})$, therefore $\lim_{n \to \infty} np = 0$. Therefore, the probability that no node fails approaches 1 asymptotically, and reliable broadcast is trivially ensured w.h.p. even with the minimum transmission range of 1.

Case 2: $p = \Omega(\frac{1}{n})$ We define a term called *quarter-neighborhood* of a node (x, y), and denote it by $qnbd(x, y)$. We associate eight quarter-neighborhoods with each node: $qnbd_A$, $qnbd_B$, $qnbd_C$, $qnbd_D$, $qnbd_{A'}$, $qnbd_{B'}$, $qnbd_{C'}$, $qnbd_{D'}$. The quarter-neighborhoods for a node (a, b) are the regions depicted in Figs. 9.2 and 9.3, and their spatial extents are tabulated in Table 9.1. Observe that $qnbd_B(a, b) = qnbd'_A(a - r - 1, b)$, $qnbd_C(a, b) = qnbd_A(a-r, b+r+1)$, and $qnbd_D(a, b) = qnbd'_A(a, b+r)$. Similarly, $qnbd_{B'}(a, b) = qnbd_A(a-r, b)$, $qnbd_{C'}(a, b) = qnbd_{A'}(a - r - 1, b + r)$, and $qnbd_{D'}(a, b) = qnbd_A(a, b + r + 1)$ Thus, if we simply consider $qnbd_A(u)$ and $qnbd_{A'}(u)$ for all nodes u, we will have considered all quarter-neighborhoods, i.e., the number of distinct (but *not disjoint*) quarter-neighborhoods is $2n$. Henceforth, we shall sometimes use $Q(x, y)$ to refer to $qnbd_A(x, y)$, and $Q'(x, y)$ to refer to $qnbd_{A'}(x, y)$. The population of each quarter-neighborhood is $r(r + 1)$. Since $d = 4r^2 + 4r = 4r(r + 1)$ in the L_∞ metric, the population of each quarter-neighborhood is $\frac{d}{4}$. We now state and prove the following result which is crucial to proving our sufficient condition for reliable broadcast:

Lemma 44. *If $p < \frac{1}{2}$ and $d \geq \max\{d_{min}, 16\frac{\ln n}{\ln \frac{1}{2p}+\ln \frac{1}{2(1-p)}}\} = \max\{d_{min}, 8\frac{\ln n}{D(Q_{\frac{1}{2}}||P)})\}$, then:*

$$\lim_{n\to\infty} \Pr[\ \forall (x,y) \ less\ than\ \frac{d}{8}\ faults\ in\ Q(x,y)\ and\ Q'(x,y)] = 1$$

Proof. As shown above, the population of any quarter-neighborhood is $\frac{d}{4}$. Each node may fail independently with probability p. Let $Y_{(x,y)}$ be a random variable denoting the number of faulty nodes in $Q(x,y)$. Then $Y_{(x,y)} = \sum_{j\in Q(x,y)} I_j$ were I_j is an indicator variable which is 1 if neighbor i of the node at (x,y) is faulty, and is 0 otherwise. $E[Y_{(x,y)}] = \frac{pd}{4}$.

Noting that $p < \frac{1}{2}$, we can apply the relative entropy form of the Chernoff-Hoeffding bound (Lemma 54) to $Y_{(x,y)}$. Observe that $d \geq \max\{d_{min}, 16\frac{\ln n}{\ln \frac{1}{2p}+\ln \frac{1}{2(1-p)}}\} \geq 16\frac{\ln n}{\ln \frac{1}{2p}+\ln \frac{1}{2(1-p)}}$. Thus, we obtain:

$$\Pr[Y_{(x,y)} \geq \frac{d}{8}] \leq e^{-\frac{d}{4}(\frac{1}{2}\ln \frac{1}{2p}+\frac{1}{2}\ln \frac{1}{2(1-p)})} \leq e^{-(\frac{16\ln n}{4(\ln \frac{1}{2p}+\ln \frac{1}{2(1-p)})}))(\frac{1}{2}\ln \frac{1}{2p}+\frac{1}{2}\ln \frac{1}{2(1-p)})} = e^{-2\ln n} = \frac{1}{n^2}$$
$$(9.7)$$

Similarly, setting $Y'_{(x,y)}$ be a random variable denoting the number of faulty nodes in $Q'(x,y)$, and following the same argument as above, we obtain that:

$$\Pr[Y'_{(x,y)} \geq \frac{d}{8}] \leq \frac{1}{n^2} \qquad (9.8)$$

By application of union bound over all $2n$ distinct quarter-neighborhoods:

$$\therefore \Pr[\forall (x,y), Y(x,y) < \frac{d}{8}\ \text{and}\ Y'(x,y) < \frac{d}{8}] \geq 1 - 2n\left(\frac{1}{n^2}\right) = 1 - \frac{2}{n}$$
$$\therefore \lim_{n\to\infty} \Pr[\forall (x,y), Y(x,y) < \frac{d}{8}\ \text{and}\ Y'(x,y) < \frac{d}{8}] = 1 \qquad (9.9)$$

$\square$

We now consider a simple broadcast protocol that is similar to the protocol that was described in [57] for the locally bounded model:

- Initially, the source does a local broadcast of the message.

- Each neighbor i of the source immediately commits to the the first value v it heard from the source, and then locally broadcasts it once in a $COMMITTED(i,v)$ message.

- Hereafter, the following protocol is followed by each node $j \notin nbd(s)$:

216

If $\frac{1}{2}r(r+1) + 1 = \frac{d}{8} + 1$ $COMMITTED(i,v)$ message are received for a certain value v, from neighbors i all lying within a single quarter-neighborhood, and not already committed to some value, commit to v, and locally broadcast a $COMMITTED(j,v)$ message.[2]

Theorem 25. *The probability that a non-faulty node shall commit to a wrong value by following the above protocol tends to 0 as $n \to \infty$.*

Proof. If all $Q(x,y)$ and $Q'(x,y)$ have strictly less than $\frac{d}{8}$ faults, the correctness of the protocol proceeds as follows:

By the assumptions of *reliable local broadcast*, if s sends exactly one message, fault-free nodes in $nbd(s)$ are guaranteed to receive it correctly. If s is faulty and sends more than one version of the message, fault-free nodes in $nbd(s)$ receive both messages, and select the first one. Thus fault-free nodes in $nbd(s)$ are guaranteed to commit to the correct value.

The rest of the proof is by contradiction. Consider the first fault-free node, say j, that makes a wrong decision to commit to a value v. From our previous assertion, j cannot be in $nbd(s)$, and hence followed protocol rules for nodes that are not s's neighbors. This implies that $\frac{d}{8}+1$ of its neighbors within some quarter-neighborhood must have broadcast a $COMMITTED$ message for v (the $COMMITTED$ messages were directly heard, leaving no place for doubt). All of these nodes cannot be faulty, as less than $\frac{d}{8}$ nodes in any quarter-neighborhood are faulty. Thus, there was at least one fault-free node that committed to v. Since j is the first fault-free node to make a wrong decision, none of the fault-free nodes amongst the $\frac{d}{8} + 1$ nodes could have made a wrong decision. Therefore, v must indeed be the correct value.

From Lemma 44, all the quarter-neighborhoods have less than $\frac{d}{8}$ faults with a probability that tends to 1 as $n \to \infty$, and hence the protocol also functions correctly with a probability that tends to 1 as $n \to \infty$. $\square$

Theorem 26. *Each non-faulty node is eventually able to commit to the correct value w.h.p.*

Proof. The proof is by induction.

---

[2]Note that $\frac{d}{8} = \frac{r(r+1)}{2}$ is always an integer, since r is assumed to take only integer values in the grid network case.

217
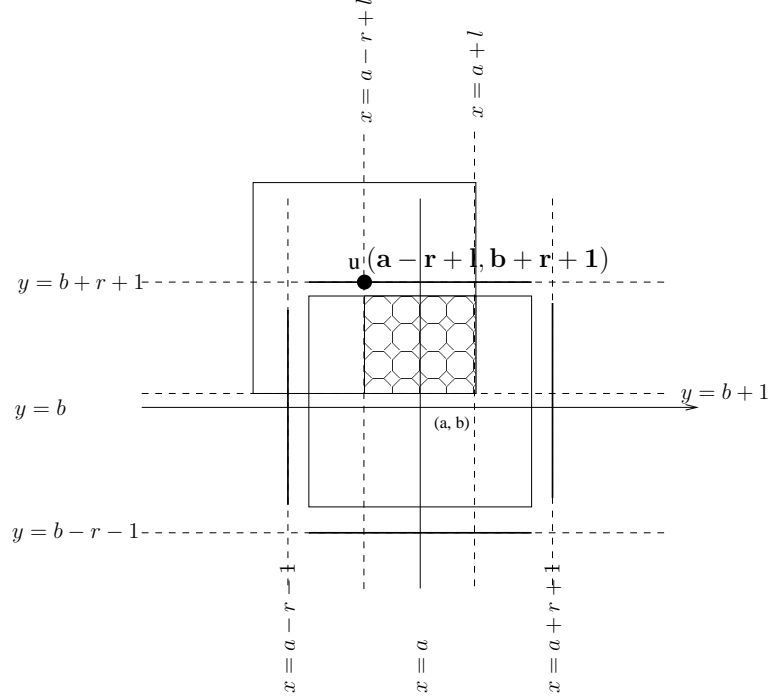
Figure 9.4: Node u has a quarter-neighborhood contained in $nbd(a, b)$

Base Case: All non-faulty nodes in $nbd(0, 0)$ are able to commit to the correct value. This follows trivially since they hear the source directly, and by assumption address-spoofing is impossible.

Inductive Hypothesis: If all non-faulty neighbors of a node located at (a, b) i.e. all non-faulty nodes in $nbd(a, b)$ are able to commit to the correct value, then all non-faulty nodes in $pnbd(a, b)$ are able to commit to the correct value.

Proof of Inductive Hypothesis: We show that each node u in $pnbd(a, b) \setminus nbd(a, b)$ has at least one of $qnbd_A(u)$, $qnbd_B(u)$, $qnbd_C(u)$, $qnbd_D(u)$, $qnbd_{A'}(u)$, $qnbd_{B'}(u)$, $qnbd_{C'}(u)$, $qnbd_{D'}(u)$ fully contained in $nbd(a, b)$. Since the population of each quarter-neighborhood is $\frac{d}{4}$, and strictly less than $\frac{d}{8}$ of the nodes in a quarter-neighborhood are faulty with probability that tends to 1 asymptotically, the number of non-faulty nodes in each quarter-neighborhood is at least $\frac{d}{8} + 1$ (since $\frac{d}{8}$ is always an integer). This ensures that the node will become aware of $\frac{d}{8} + 1$ nodes in $nbd(a, b)$ having committed to a (the correct) value, and will also commit to it (if it is non-faulty). The situation is depicted in Fig. 9.4 for $u \in \{(a - r + l, b + r + 1) | 1 \leq l \leq r\}$, for which $qnbd_A(u)$ lies in $nbd(a, b)$. For other locations, a similar argument holds. □

218

9.6 Byzantine Failures in a Random Network: Sufficient Condition

We obtain a sufficient condition for a network of n nodes deployed uniformly at random, based on the sufficient condition for the grid network model. To maintain consistency with the grid network formulation, we again assume a toroidal region of area $\sqrt{n}$ x $\sqrt{n}$, with n nodes located uniformly at random. The average degree of a node is the average number of the remaining $n-1$ nodes that fall within its neighborhood. Recall that we are using L_∞ distance metric), and thus the average degree is $d_{avg}(n,p) = \frac{(n-1)(2r(n,p))^2}{n} = 4r^2(n,p)(1 - \frac{1}{n}) \approx 4r^2(n,p)$ for large n.

Theorem 27. *Assuming the L_∞ metric, in a random network with Byzantine failure probability $p < \frac{1}{2}$, and $r(n,p) \geq \sqrt{\frac{100 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}}}$:*

$$\lim_{n \to \infty} \Pr[\text{ reliable broadcast succeeds }] = 1$$

Proof. We begin with the observation that if $r(n,p)$ becomes so large that a node's range spans the entire network, all nodes are neighbors, and trivially broadcast is achievable. Thus, this result is of interest only so long as $r(n,p)$ is not so large.

In light of Fact 1:

$$
\begin{aligned}
D(Q_{\frac{1}{2}} \| p) &= \frac{1}{2} \ln \frac{1}{2p} + \frac{1}{2} \ln \frac{1}{2(1-p)} \\
&\geq \frac{1}{2}(1 - 2p) + \frac{1}{2} \ln \frac{1}{2(1-p)} \\
&= \frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}
\end{aligned}
\tag{9.10}
$$

Also, since $p < \frac{1}{2}$:

$$
0 < \frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)} \leq \frac{1}{2}(1 - \ln 2) < 1
\tag{9.11}
$$

Similar to grid networks, we use a notion of quarter-neighborhoods. For a given broadcast instance, we again use relative coordinates by treating the source's coordinates as $(0,0)$. With some abuse of the grid network notation introduced in Section 9.1, we can extend the notion of $nbd(x,y)$, to include all nodes within distance r of point (x,y) (regardless of

whether or not there is a node at (x, y)), where x and y are real numbers. The notion of $pnd(x, y)$ is also similarly extended to all points (x, y).

Note that in this model, a node's (or point's) coordinates are real numbers. We thus associate eight quarter-neighborhoods with each node, with spatial extents as in Table 9.1, except that now x and y must be treated as real numbers. Also, now it is not possible to assert that there are only $2n$ distinct quarter-neighborhoods. Thus, all eight quarter-neighborhoods of a node must be treated as distinct[3], yielding $8n$ quarter-neighborhoods in all.

The quarter-neigborhoods are axis-parallel rectangles of area $r(n, p)(r(n, p) - 1) \geq \frac{r^2(n,p)}{2}$ (for $r(n, p) \geq 2$). Then, if $r^2(n, p) \geq \frac{100 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}}$, then we can apply Lemma 58 for all axis-parallel rectangles of area $r(n, p)(r(n, p) - 1) \geq \frac{50 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}} \geq \frac{100 \ln n}{1 - \ln 2}$, to obtain that they all have at least $\frac{50 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}} - 50 \ln n > \frac{25 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}} > \frac{50 \ln n}{1 - \ln 2}$ nodes, with probability at least $1 - \frac{50 \ln n}{n} \to 1$.

Thus all such rectangles are *non-empty*. Also:

$$\frac{25 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}} \geq \frac{25 \ln n}{D(Q_{\frac{1}{2}} || p)} > \frac{8 \ln n}{D(Q_{\frac{1}{2}} || p)} \tag{9.12}$$

Hence, all the quarter-neighborhoods have at least $\frac{8 \ln n}{D(Q_{\frac{1}{2}} || p)}$ nodes (which is the quarter-neighborhood population in the grid network case). Then using a proof argument similar to Lemma 44, one can prove the following result:

Lemma 45. *If $p < \frac{1}{2}$, and $r(n, p) \geq \sqrt{\frac{100 \ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}}}$, then*

$$\lim_{n \to \infty} \Pr[\text{ all } 8n \text{ qnbds have non-faulty majority }] = 1$$

In light of this, one can use a broadcast protocol similar to that for grid networks (a node commits to a value if it is received from a majority of the nodes in some quarter-neighborhood), and, for all broadcast sources, and instances, the reliable broadcast properties continue to hold, as follows:

Relying on Lemma 45, we can apply a proof argument similar to Theorem 25 to argue that with high probability no non-faulty node will commit to a wrong value.

We can also show that each non-faulty node will eventually be able to commit to the

---

[3]Note that distinct does not mean disjoint.

correct value w.h.p. The proof is by induction, similar to the proof of Theorem 26, except that the terms $nbd(x,y), pnd(x,y)$ must be interpreted as per their re-definition in this section (i.e., the region within distance r of a point (x,y), regardless of whether there is a node at that point).

In the base case, all neighbors of the source (which is at $(0,0)$) commit to the correct value trivially. In the inductive step, one can show that if all nodes in $nbd(x,y)$ (as per the re-defined notation) have comitted to the correct value, all nodes in the region $pnd(x,y) \setminus nbd(x,y)$ have some quarter-neighborhood contained in $nbd(x,y)$, and can commit to the value received from a majority of nodes in this quarter-neighborhood. $\square$

Since the area within range of a node is at most $4r^2$ (for the valid domain of r values) in the L_∞ metric, the result indicates that an average node degree d_{avg} of $\frac{400 \ln n}{\frac{1}{2}-p+\frac{1}{2} \ln \frac{1}{2(1-p)}}$ suffices for reliable broadcast. Hence the *critical* average node degree $d^{avg}_{critical}$ is $O(\frac{\ln n}{\frac{1}{2}-p+\frac{1}{2} \ln \frac{1}{2(1-p)}})$. A more intuitive way of viewing the result is that *critical* average degree in a random network is $O(\max\{\ln n, \frac{\ln n}{D(Q_{\frac{1}{2}}||P)}\})$ or $O(\ln n + \frac{\ln n}{D(Q_{\frac{1}{2}}||P)})$.

9.7 Crash-Stop Failures in a Grid Network

We now consider the achievability of reliable broadcast in a grid network when nodes may cease to function with probability p. This is equivalent to the network being connected despite failures. Our results for this scenario improve upon prior results by Shakkottai et al., in [101].

Theorem 28. *In a grid network where nodes can exhibit crash-stop failure with probability* $p \leq 1 - \frac{1}{\ln n}$, *if* $r(n,p) < \max\{1, \frac{1}{4}\sqrt{\frac{\ln n}{\ln \frac{1}{p}}}\}$:

$$\lim_{n\to\infty} \Pr[\ disconnection\ ] = 1$$

Proof. Evidently the minimum transmission range required for connectivity is at least 1, corresponding to $d = d_{min} = 8$ (in L_∞ metric), else the degree of all nodes is 0 (except in the case when all nodes are faulty, and connectivity becomes irrelevant). Thus, we only focus on the case where $\frac{1}{4}\sqrt{\frac{\ln n}{\ln \frac{1}{p}}} > 1$. In this scenario $r(n,p) < \max\{1, \frac{1}{4}\sqrt{\frac{\ln n}{\ln \frac{1}{p}}}\}$ implies that $r(n,p) < \frac{1}{4}\sqrt{\frac{\ln n}{\ln \frac{1}{p}}}$.

We show that when $p \leq 1 - \frac{1}{\ln n}$, the network is asymptotically disconnected with probability approaching 1 if $r < \frac{1}{4}\sqrt{\frac{\ln n}{\ln \frac{1}{p}}}$.

In the L_∞ metric, having $r(n,p) < \frac{1}{4}\sqrt{\frac{\ln n}{\ln \frac{1}{p}}}$ yields a node degree $d(n,p) = 4r^2 + 4r \leq 8r^2 < \frac{\ln n}{2\ln \frac{1}{p}}$.

Consider a particular node j in the network. If j is non-faulty, but all its neighbors are faulty, we have a potential disconnection event. Given that there are d neighbors, and each may fail independently with probability p, the probability that j does not fail, but all nodes in $nbd(j)$ fail, is $(1-p)p^d$.

Since $p \leq 1 - \frac{1}{\ln n}$, we obtain that:

$$1 - p \geq \frac{1}{\ln n} \tag{9.13}$$

$$\Pr[\text{ A given node } j \text{ is non-faulty, but isolated}]$$
$$= \Pr[j \text{ is non-faulty and all neighbors of } j \text{ are faulty }]$$
$$= (1-p)p^d \geq \left(\frac{1}{\ln n}\right)p^{\frac{\ln n}{2\ln \frac{1}{p}}} = \left(\frac{1}{\ln n}\right)\left(\frac{1}{\sqrt{n}}\right) = \frac{1}{\sqrt{n}\ln n} \tag{9.14}$$
$$\geq \frac{(\ln n)^3}{n}\text{for large } n$$

Note the following:

$$d < \frac{\ln n}{2\ln \frac{1}{p}} \leq \frac{\ln n}{2(1-p)} \leq \frac{(\ln n)^2}{2} \quad \text{(Fact 1, (9.13))} \tag{9.15}$$

Let us mark out a subset of nodes j such that the neighborhoods of these nodes are all disjoint, as in Fig. 9.1. Then, from Fact 3, the number of such nodes that we may obtain is at least $\frac{n}{2d}$ for large n.

Let I_j be an indicator variable that takes value 1 if j is non-faulty but isolated. Then $Pr[I_j = 1] \geq \frac{(\ln n)^3}{n}$, and all I_j's are i.i.d.

Let X be a random variable denoting the number of nodes from the chosen set that are non-faulty and isolated. Then $X = \sum I_j$, and $E[X] \geq \frac{(\ln n)^3}{n}(\frac{n}{2d}) \geq \frac{(\ln n)^3}{(\ln n)^2} = \ln n$. We can thus set $\beta = \frac{1}{2}$ in the Chernoff bound of Lemma 53, and obtain that:

$$\Pr[X > \frac{\ln n}{2}] \geq 1 - e^{-\frac{\ln n}{8}} = 1 - \frac{1}{n^{\frac{1}{8}}} \tag{9.16}$$

Thus, for $p < 1 - \frac{1}{\ln n}$ and large n:

$$\lim_{n \to \infty} \Pr[\text{ At least two non-faulty nodes are isolated}] = 1 \qquad (9.17)$$

Hence a broadcast from one such node will not be received by the other node. □

We also briefly touch upon the range of p values satisfying $1 - p = o\left(\frac{1}{n}\right)$. By applying the union bound, the probability that at least one node is non-faulty is at most $n(1-p)$. Since $1 - p = o\left(\frac{1}{n}\right)$, we know that $\lim_{n \to \infty} n(1-p) = 0$. Therefore:

$$\lim_{n \to \infty} \Pr[\text{ all nodes are faulty }] = 1 \qquad (9.18)$$

Thus the issue of connectivity is irrelevant.

We now present a sufficient condition for the asymptotic connectivity.

Theorem 29. *In a grid network with crash-stop failure probability $p < 1$, when $r(n,p) \geq \max\{1, \sqrt{\frac{8 \ln n}{\ln \frac{1}{p}}}\}$:*

$$\lim_{n \to \infty} \Pr[\text{ the network is connected }] = 1$$

Proof. $p = o(\frac{1}{n})$

When the failure probability is so small as to fall in this range, by applying the the union bound, we obtain that the probability of even a single node failing is at most np. Since, $\lim_{n \to \infty} np = 0$, asymptotic connectivity is trivially ensured even with the minimum transmission range of 1.

$p = \Omega(\frac{1}{n})$

Note that when p is $\Omega(\frac{1}{n})$, then $r(n,p) > 1$ for large enough n. Consider the subdivision of the grid as depicted in Fig. 9.5, so that the resulting cells have x-extents (and also y-extents) 0 to a, $a+1$ to $a+b$, $a+b+1$ to $2a+b+1$, $2a+b+2$ to $2a+2b+1$, and so on, where $a = \lfloor \frac{r}{2} \rfloor$ and $b = r - a = r - \lfloor \frac{r}{2} \rfloor$. It is easy to see that each node is within range of all other nodes in the cells adjoining its own (as depicted in Fig. 9.5). If each cell has at least one non-faulty node, there exists a connected backbone that covers all points, and hence all nodes. Therefore, all non-faulty nodes are connected to each other via this backbone. The populations of the cells thus obtained can be $(a+1)^2$, $(a+1)b$ or b^2. Since $a+1 = \lfloor \frac{r}{2} \rfloor + 1 \geq \frac{r}{2}$, and $b = r - \lfloor \frac{r}{2} \rfloor \geq \frac{r}{2}$, the population k of any cell satisfies $k \geq \frac{r^2}{4}$, and
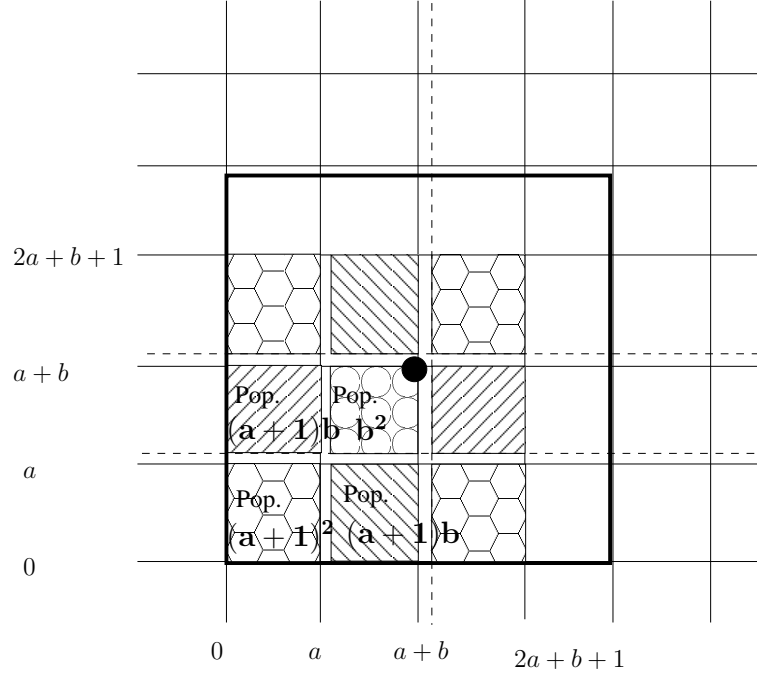
Figure 9.5: Subdivision of network into cells (all adjacent cells are within range)

the maximum possible number of cells $m \leq \frac{4n}{r^2}$. Then:

$$\Pr[\text{ no non-faulty node in a given cell }] = p^k \leq p^{\frac{r^2}{4}} \tag{9.19}$$

Since $r \geq \sqrt{\frac{8 \ln n}{\ln \frac{1}{p}}}$:

$$\Pr[\text{ no non-faulty node in a given cell }] \leq p^{\frac{r^2}{4}} \leq p^{\frac{2 \ln n}{\ln \frac{1}{p}}} = e^{-2 \ln n} = \frac{1}{n^2} \tag{9.20}$$

The total number of cells is at most $\frac{4n}{r^2} \leq 4n$ since $r \geq 1$ (however note that $\frac{4n}{r^2}$ is actually less than n for large enough n, whenever $p = \Omega(\frac{1}{n})$). Applying a union bound over all cells:

$$\Pr[\text{ at least 1 non-faulty node in each cell }] \geq 1 - \frac{4}{n} \tag{9.21}$$

Since this condition ensures connectivity, we obtain that:

$$\lim_{n \to \infty} \Pr[\text{ network is connected }] = 1 \tag{9.22}$$

$\square$

224
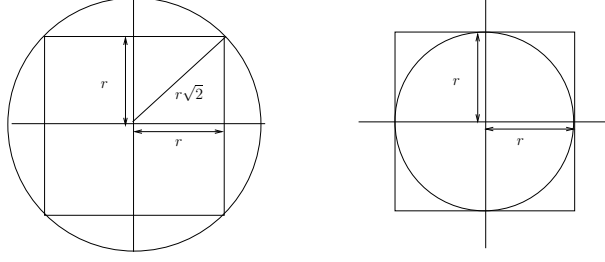
Figure 9.6: Relationship between L_∞ and L_2 neighborhoods

9.8 Conditions in Euclidean Metric

We show that our results derived for L_∞ metric continue to hold for L_2 metric, with only the constants in the theta notation changing.

Lemma 46. *If reliable broadcast is achievable asymptotically in L_∞ for all $r \geq r_{min}$, then it is achievable asymptotically in L_2 for all $r \geq \sqrt{2}r_{min}$.*

Proof. The proof is by contradiction. Suppose that, for a given failure configuration, broadcast is asymptotically achievable in L_∞ for all $r \geq r_{min}$ but is not asymptotically achievable for all $r \geq \sqrt{2}r_{min}$ in L_2. Observe that it is possible to circumscribe a L_∞ neighborhood of range r by a L_2 neighborhood of range $\sqrt{2}r$ (Fig. 9.6). Hence the non-faulty nodes in an L_2 network of transmission range $\sqrt{2}r$ can be made to simulate the operation of nodes in a L_∞ network with range r (as the L_∞ neighborhood is fully contained within the L_2 neighborhood). Also, given that all nodes in the L_2 network know the locations of their neighbors, and no address spoofing is allowed, the faulty nodes (in the Byzantine failure case) cannot gain any unfair advantage by not simulating the the L_∞ network. If there is some $r \geq r_{min}$ for which we can achieve broadcast in the L_∞ network asymptotically, but not in the the L_2 network of range $\sqrt{2}r$, we obtain a contradiction, as achievability in the L_∞ network would imply achievability in the L_2 network. This implies that if broadcast is achievable in the L_∞ network of range r , so must it be in the L_2 network of range $\sqrt{2}r$. □

Lemma 47. *If reliable broadcast fails asymptotically in L_∞ for all $r \leq r_{min}$, then it fails asymptotically in L_2 for all $r \leq r_{min}$.*

Proof. The proof is by contradiction. Suppose that broadcast fails asymptotically in L_∞ for range r, but does not fail in L_2 for range r. Observe that an L_∞ neighborhood of transmission range r circumscribes an L_2 neighborhood of range r (Fig. 9.6). Thus, for any

given failure configuration, if broadcast succeeds in the the L_2 network of range r, so can it in the L_∞ network of radius r, as we could simply make the fault-free nodes in the L_∞ network simulate the behavior of nodes in the L_2 network. Hence, if broadcast does not fail in the L_2 network of range $r \le r_{min}$, it will not fail in the L_∞ network of range $r \le r_{min}$. This yields a contradiction. □

9.9   Non-Toroidal Networks

We used the assumption that the network is toroidal to avoid edge effects. However, it can be seen that the results (in terms of transmission range $r(n, p)$) would continue to hold even if the network were spread over a non-toroidal rectilinear domain.[4]

The necessary condition would continue to hold, since the area within transmission range at the edges can be no more more than the area within transmission range (and hence degree) of nodes towards the center, and if reliable broadcast is not achievable for a certain value of $r(n, p)$ even with the assumption that all nodes have equal network area within their transmission range, then it must certainly be impossible when some nodes (those near the edges of the network region) have a smaller area within range.

The sufficient conditions for Byzantine failures continue to hold since the described algorithms rely on information from quarter-neighborhoods, and it can be seen that even the nodes at the edges have at least one quarter-neighborhood within the network region. Hence, if some value of $r(n, p)$ suffices in a toroidal network, the same would suffice in the corresponding non-toroidal network as well. For the crash-stop failure case, the sufficient condition continues to hold as even nodes at the edges have at least one full cell within their range.

9.10   Discussion

An interesting observation is that the form of the results for Byzantine failures is very similar to the results for crash-stop failures/connectivity. For Byzantine failures, we have obtained that the critical node degree for grid networks is $\Theta(d_{min} + \frac{\ln n}{\ln \frac{1}{2p} + \ln \frac{1}{2(1-p)}})$, which may be re-stated as $\Theta(d_{min} + \frac{\ln n}{D(Q_{\frac{1}{2}} || P)})$ where $Q_{\frac{1}{2}}$ denotes the $Bernoulli(\frac{1}{2})$ distribution,

---

[4]Note that the degree in a non-toroidal network is a function of node location; hence it is more relevant to state results in terms of transmission range $r(n, p)$ instead of degree.

P denotes the *Bernoulli*(p) distribution, and $D(Q||P)$ denotes the *relative entropy* (or Kullback-Leibler distance) between distributions Q and P. Similarly, the node degree for crash-stop failures/connectivity is $\Theta(d_{min} + \frac{\ln n}{\ln \frac{1}{p}})$, and may be viewed as as $\Theta(d_{min} + \frac{\ln n}{D(Q_1||P)})$, where Q_1 is the *Bernoulli*(1) distribution, and P is the *Bernoulli*(p) distribution (using the standard convention that $0 \ln \frac{0}{1-p} = 0$ where $1 - p > 0$ is a valid probability value). These results have a similar structural form, involving a minimum term required for connectivity without faulty behavior, and a second term required to ensure broadcast even in presence of failure.

Recall that we derive the necessary condition from isolated failure events, and this is found to match the sufficient condition within a constant factor. Thus, it is possible that failure events involving isolated nodes not determining the correct broadcast value may be the dominant failure events [5].

Focusing on these isolated failure events, the obtained expressions for node degree can be explained in the light of Sanov's Theorem [24]. As per Sanov's Theorem, the probability of occurrence of the event-set $\mathcal{E} = \{$ half or more neighbors faulty$\}$ is dominated by the probability of the event in $\mathcal{E}$ closest in relative entropy to the governing fault distribution P. Since we are considering the regime $p < \frac{1}{2}$, the closest event is that of exactly half the neighbors being faulty, corresponding to $Q_{\frac{1}{2}}$. In light of this, the critical degree expression for Byzantine failures is quite intuitive. One can similarly explain the crash-stop results.

The necessary and sufficient condition for connectivity in a sensor network where nodes sleep with probability p was shown in [55] to be $\Theta(\frac{\ln(n(1-p))}{1-p})$ (when expressed in our notation) for the case of a randomly deployed network. This problem is equivalent to that of crash-stop failures in random networks. Our sufficient condition for random networks with Byzantine failure probability $p < \frac{1}{2}$ is $O(\frac{\ln n}{\frac{1}{2} - p + \frac{1}{2} \ln \frac{1}{2(1-p)}})$.

There is a similarity of form in the two results, and one may interpret the critical node degree as being $O(\ln n(1 - p) + \frac{\ln n(1-p)}{D(Q||P)})$ where Q is the *Bernoulli*(q) distrbution, and P is the *Bernoulli*(p) distribution; $q = 1$ (and $p < 1$) for the sleeping/crash-stop case in [55], and $q = \frac{1}{2}$ (with $p < \frac{1}{2}$) for the Byzantine failure case.

Additionally, it is evident that our expressions for the grid network and random network

---

[5]Note that in [42], it was found that the primary disconnection events in non-faulty *random* networks are those involving single isolated nodes.

diverge when $p \to 0$, but are otherwise within a constant factor of each other (for p bounded away from 0). This difference is quite intuitive. In a grid network, as failure probability $p \to 0$, the network tends towards a deterministic topology, whereas in a random network, if failure or sleep probability $p \to 0$, the network can only tend towards a denser but still random network. Thus, at small values of p, a very small degree will suffice for a grid network, but may not for a random network. At larger p values, the grid network exhibits increasing randomness and begins to resemble a network with random deployment. Thus, one may see that the two expressions are within constant factor of each other when p is large (given sufficiently large n), but diverge as $p \to 0$.

Chapter 10

Reliable Local Broadcast with Byzantine Failures

In Chapter 7, we briefly reviewed results in the literature on achieving reliable broadcast in wireless networks. In Chapter 8 and Chapter 9, we described results for achievability of reliable broadcast under different assumptions regarding the network and fault model. Our results in these previous chapters, as well as a substantial amount of the prior work reviewed in Chapter 7, assumes that if a node transmits a message it is received by each and every node within a designated neighborhood in its spatial vicinity. This eliminates the potential for *duplicity* by a Byzantine source node, and ensures local agreement. While this model reflects the shared nature of the wireless medium, it fails to capture its unreliability. The wireless medium can be extremely unreliable, and can show highly variable channel quality over time, due to multipath effects. This can lead to significant fluctuation in the received signal. Resultantly, there is often a non-negligible probability of unsuccessful reception, even in the absence of malicious collision-causing behavior. Thus, any attempt at designing reliable broadcast protocols based on these theoretical results must begin with an effort to implement a *reliable local broadcast* primitive in a *scalable* manner.

One might envision implementing local broadcast by running a point-to-point Byzantine agreement protocol, with retransmissions over every lossy (point-to-point) link to handle channel errors. However, such a solution may not be scalable, as the underlying medium is shared and thus the operation of nearby (point-to-point) links cannot occur concurrently, and must be serialized.

While the issues of reliable broadcast and consensus in the presence of a bounded number of collisions/spoofing have been addressed in recent years, such as [58] and [38], probabilistic channel losses have typically not been considered. Random transient Byzantine failures that include collision-causing is examined in [91]. Though also of a probabilistic nature, their model is different in that nodes either fail to transmit, transmit a wrong value or transmit

out of turn, with a certain probability, in each round.

In this chapter, we investigate the possibility of designing Byzantine fault-tolerant communication primitives that can work in the presence of channel unreliability. We continue to assume that the physical(PHY) and medium-access control (MAC) layers are fault-free (i.e., nodes do not deliberately cause collision or spoof MAC addresses). *Our primary intent is to highlight the potential for lightweight scalable solutions that exploit knowledge of physical layer characteristics, in conjunction with other information provided by lower layers, to achieve message-ordering conditions useful for reliable communication.* We sketch out a simple proof-of-concept algorithm that can facilitate the implementation reliable local broadcast with probabilistic guarantees in a *local broadcast domain.* We also briefly discuss how the proposed reliable local broadcast solution can be optimized further, and also be used as a sub-protocol in a global broadcast algorithm for multi-hop networks.

A preliminary version of the work described in this chapter was reported in [10].

10.1 How a Lossy Wireless Channel Inhibits Reliable Local Broadcast

In this section we briefly discuss how an unreliable wireless channel can affect the achievability of reliable local broadcast.

Consider a source s that originates a message, which needs to be locally broadcast to its neighbors. However, as the channel is lossy, each neighbor successfully receives the message only with a certain probability. Resultantly, it is possible that a transmission may only be heard by some subset of s's neighbors. If s is non-faulty, this issue can be readily resolved by having s retransmit the message a sufficient number of times to ensure that each neighbor receives at least one copy with high probability (w.h.p.). However consider what might transpire if s is faulty, and seeks to leverage the channel's unreliability to create confusion amongst its neighbors:

Suppose that s initially sends a message m with value 0. Some of its neighbors do not receive it, i.e., it is received by some subset $\mathcal{N}_1$ of s's neighbors. It then sends another version of the same message, containing a value 1. This message is received by some subset $\mathcal{N}_2$. If $\mathcal{N}_1 \setminus \mathcal{N}_2$ is non-empty, there are certain nodes that will assume that s sent only one value, i.e., 0. If $\mathcal{N}_2 \setminus \mathcal{N}_1$ is non-empty, there are certain nodes that will assume that s

sent only one value, i.e., 1. Nodes in $\mathcal{N}_1 \cap \mathcal{N}_2$ receive both values, and are in a position to detect s's duplicity. These nodes can choose a default value, e.g., the first value sent by s. However, there still remains the issue of ensuring that the other nodes do the same. One approach might consist in the raising of an alarm by nodes in $\mathcal{N}_1 \cap \mathcal{N}_2$, but would require a means for the other nodes to resolve whether the alarm(s) are to be trusted. Another possible approach involves using a point-to-point Byzantine agreement algorithm in the neighborhood of s. However, these approaches have high message-overhead. In particular, given the shared nature of the wireless medium, the messages must be sent in turn on the same medium, thereby exacerbating the cost.

Thus, one may prefer to have a more lightweight approach to ensure agreement of all nodes on a common value (and potentially rely on the fact that after a number of duplicitous transmissions by s, all nodes would at some time detect its duplicity themselves, and s would be universally identified as untrustworthy).

10.2   Causal Ordering and Physical Clocks

In this section, we briefly review notions of clocks and ordering that are relevant to the discussion in this chapter.

We assume the existence of some frame of reference external to the system. The physical time in this frame of reference is considered to be an absolute measure of physical time for the purpose of our discussion. Thus, at time instant t, the external clock value is t.

Each node u in the system has its own physical clock. The clock value of a node u at time instant t is denoted by $C_u(t)$. When we refer to external synchronization within bound D, we imply synchronization to this ideal external clock within bound D, i.e., at each time instant t: $|C_u(t) - t| \leq D$.

Clock drift is modeled as being linear, i.e., if the true elapsed time is T, the observed elapsed time lies in the range $[(1 - \delta)T, (1 + \delta)T]$, where δ is the drift per unit time (also referred to as drift-rate).

When we refer to internal synchronization within bound D, we imply that at any time instant t, the clocks of two internally synchronized nodes u, w satisfy: $|C_u(t) - C_w(t)| \leq D$. When we refer to a node adjusting its clock, we imply that the node applies a correction to its clock value.

In his seminal paper [69], Lamport proposed that a key goal in a distributed system should be to ensure that causal relationships are respected. This causality could be captured in a *happened-before* relation, which imposes a partial order on system events. Thus, $a \rightarrow b$ implies that *a happened-before b*, and b may be causally affected by a. Let $C(a)$ denote the time observed for an event a as per a clock C. A satisfactory clock C must then satisfy the following:

Definition 4. *(Clock Condition [69]) For any events a, b: $a \rightarrow b \implies C(a) < C(b)$.*

To this effect, Lamport logical clocks were proposed in [69]. An anomalous scenario was also considered whereby out-of-system message exchanges could lead to violation of the Clock Condition. This leads to the consideration of a *Strong Clock Condition* whereby causal ordering is preserved even taking into account out-of-system messages. It was observed in [69] that if the clock drift rate δ, the maximum clock skew (or synchronization bound) D and the minimum message transmission time T_l satisfy the relation: $T_l \geq \frac{D}{1-\delta}$, then the system of *physical* clocks satisfies the Strong Clock Condition. It was also shown that a simple synchronization algorithm suffices to ensure that clock skew is bounded by a suitable D.

The notion of leveraging physical clocks rather than logical clocks has wider significance. Consider a system where some processes may exhibit Byzantine behavior. Then their logical clock values cannot be trusted, as they may affix incorrect logical clock values to messages they send, in order to taint the logical clocks of other processes. If one could ensure that the physical clocks of non-faulty nodes satisfy certain ordering conditions, this could be quite beneficial. A similar intuition underlies our approach towards reliable local broadcast.

10.3 Loose Synchronization and Local Broadcast

In this section we describe the basic assumptions and approach behind leveraging the existence of loose synchronization to facilitate a certain ordering condition between locally broadcast messages. In Section 10.4, we discuss how the ordering condition can be realized in a wireless network, and subsequently describe in Section 10.5 how it might be leveraged to achieve reliable local broadcast with probabilistic guarantees.

Consider a system comprising a node v that is interested in sending messages, and a set of

other nodes (neighbors of v) capable of receiving messages from v over a *shared* broadcast medium. Each node is equipped with a single half-duplex transceiver. Thus, no node can send and receive messages simultaneously, and only one message can be successfully transmitted or received at a time by a node. Note that this is a reasonable model for wireless nodes equipped with a single half-duplex transceiver and an omnidirectional antenna, which operate on a single common channel.

Receive-Timestamp   A node is assumed capable of noting its local physical clock value just after its physical layer finishes receiving a message (this is also a reasonable assumption; such a timestamping operation could be implemented in hardware). This is termed as the receive-timestamp observed by the node for the message.

The messages sent in this system have the following property:

The minimum (absolute) time the packet transmission occupies the channel is T_l, and the actual total (absolute) time taken by a message in transit (between the time the sending node's physical layer starts sending the message, and the time the receiving node finishes receiving and notes its receive-timestamp) is upper-bounded by T_u. Hence $T_u - T_l$ subsumes the maximum propagation delay and upper bounds on any processing delays incurred up to the time of taking the timestamp.

Therefore, the (absolute) time T taken by a message in transit from sender to receiver (between timestampings) satisfies $T_l \leq T \leq T_u$. Note that this condition is satisfied by all messages including those sent by faulty nodes. We explain in Section 10.4 why this is a reasonable assumption.

We define the following condition:

Definition 5. *(Receipt-Order Condition) If a node v sends a message m_1, followed by a message m_2, then for all non-faulty nodes u, w which are neighbors of v: the receive-timestamp observed by u for m_2 is greater than the receive-timestamp observed by w for m_1.*

We identify two situations in which the Receipt-Order Condition holds. The first one relies on assumptions about external clock synchronization, and the second one relies on assumptions about internal clock synchronization.

Observation 1. *(Externally Synchronized Nodes) If the physical clocks of all non-faulty nodes in the system are externally synchronized within bound D, and if $2T_l - T_u > 2D$, then the local physical timestamps observed by the non-faulty neighbors of v for messages sent by v satisfy the Receipt-Order Condition.*

Proof. Suppose the sender starts sending the two messages m_1, m_2 at times t_1 and t_2 respectively (according to the ideal external clock). Then those non-faulty neighbors of v that received m_1 would have received it within the interval $[t_1 + T_l, t_1 + T_u]$ (as per the external clock), and their observed receive-timestamp would lie in the range $[t_1 + T_l - D, t_1 + T_u + D]$. Similarly, the observed receive-timestamp for the second message m_2 falls within $[t_2 + T_l - D, t_2 + T_u + D]$. Since the two messages are sent by v, using its half-duplex transceiver, on the same medium, they are temporally ordered and separated in time i.e. $t_2 \geq t_1 + T_l$. Thus, $(t_2 + T_l - D) - (t_1 + T_u + D) = t_2 - t_1 - T_u + T_l - 2D \geq 2T_l - 2D - T_u > 0$. Therfore, any non-faulty node that receives the first message observes a receive-timestamp that is less than the receive-timestamp for the second message observed by those non-faulty nodes that see the second message. Hence, the Receipt-Order Condition holds. □

Observation 2. *(Internally Synchronized Nodes) Consider an interval of time in the system in which no non-faulty node adjusts its physical clock, the physical clocks of all non-faulty nodes stay internally synchronized within bound D, and drift-rate is upper-bounded by δ. We are interested in messages sent and received entirely during this interval. If $2T_l - T_u - \delta(2T_l + T_u) > D$, then the local physical timestamps observed by the non-faulty neighbors of v for messages sent by v satisfy the Receipt-Order Condition.*

Proof. The argument is almost the same as that used in [69] to argue that a system of physical clocks can be made to satisfy the *Strong Clock Condition*, except that we now apply it in the context of a broadcast medium with multiple recipients of the same message.

Denote by $E_v^s(m)$, the event of node v sending message m, and by $C_u(E_v^s(m))$ the local physical clock time at some non-faulty node u, at the time v started the transmission. Note that this does not imply that node u is aware of the instant at which transmission started. u may only detect the transmission after some minimum propagation delay. Denote by $E_u^r(m)$, the event of node u receiving message m, and by $C_u(E_u^r(m))$, the receive-timestamp observed by node u for a message m received by it (recall that receive timestamps are recorded when the reception has finished).

Suppose a node v starts sending a message m_1 at a time when local time at some non-faulty neighbor u is $C_u(E_v^s(m_1))$. Thus, from the assumption that clocks are internally synchronized within bound D, the local time at any other non-faulty neighbor w must be $C_w(E_v^s(m_1)) \leq C_u(E_v^s(m_1)) + D$, and w will observe a receive-timestamp $C_w(E_w^r(m_1)) \leq C_w(E_v^s(m_1)) + T_u(1 + \delta) \leq (C_u(E_v^s(m_1)) + D) + T_u(1 + \delta)$.

If v later starts sending a message m_2 when local-time at u is $C_u(E_v^s(m_2))$, then $C_u(E_v^s(m_2)) - C_u(E_v^s(m_1)) \geq T_l(1 - \delta)$. Thus the receive-timestamp u observes for m_2 is at least $C_u(E_u^r(m_2)) \geq C_u(E_v^s(m_2)) + T_l(1 - \delta) \geq C_u(E_v^s(m_1)) + 2T_l(1 - \delta)$. Thus, for u and any other non-faulty node w: $C_u(E_u^r(m_2)) \geq C_u(E_v^s(m_1)) + 2T_l(1-\delta) = (C_u(E_v^s(m_1)) + D + T_u(1+\delta)) - T_u(1+\delta) - D + 2T_l(1-\delta) \geq C_w(E_w^r(m_1)) + (2T_l(1-\delta) - T_u(1+\delta) - D) = C_w(E_w^r(m_1)) + (2T_l - T_u - \delta(2T_l + T_u) - D) > C_w(E_w^r(m_1))$.

Thus the Receipt-Order Condition is satisfied. $\qquad\square$

10.4 Network Model

Consider a wireless multi-hop network. The set of nodes within transmission range of a node v is termed $nbd(v)$. v is a member of $nbd(v)$. Let $nbd'(v) = nbd(v) \setminus \{v\}$.

For the purpose of our discussion, we focus on a *local broadcast domain* within the wireless network, comprising a sender node s and nodes within its transmission-range, denoted by $nbd'(s)$, to which we wish to ensure reliable local broadcast delivery. We denote $|nbd'(s)|$ by d, and define $d_o = \min_{x \in nbd'(s)} nbd'(x) \cap nbd'(s)$. Thus d_o is the minimum number of common neighbors of s and any of its neighbors.

10.4.1 Fault Model

We assume the locally bounded fault model of Chapter 8, wherein an adversary may place faults so long as the number of faults in any single neighborhood does not exceed a specified number b. Faulty nodes can exhibit Byzantine behavior at higher layers, i.e., they may change the values/semantics of messages. However all PHY/MAC layers are non-faulty and faulty nodes do not deliberately cause collisions or spoof MAC addresses.

10.4.2 Communication Model

We allow for an unreliable wireless channel where fading and other effects may lead to non-ideal transmission characteristics. Accidental collisions and interference are possible, due to an imperfect medium access mechanism. If a node transmits a message, the probability that a neighbor successfully receives it is p_s. Packet errors due to fading, or accidental interference etc. are subsumed in the error probability $(1-p_s)$. The probability of successful reception p_s is assumed independent and identical for each transmission and each receiving node. A desired access probability $0 < p_a < 1$, and an accordingly large enough timeout T_a are chosen, such that if a packet was put into a node's outgoing queue at time t, then by time $t + T_a$, the packet gets a chance to be transmitted by this node and received by neighbors with probability at least p_a (assumed to be independent of other nodes for simplicity). Both p_s and p_a are assumed independent of d, d_o. Note that T_a is a function of the target access probability p_a, as well as the lengths of packet-queues (and hence traffic-levels in the network).

All nodes possess a single half-duplex transceiver with an omnidirectional antenna, and operate on a single channel. They also use a single transmission rate [1], and all valid messages are of a predetermined (and equal) size (as discussed later, this can be chosen to facilitate reliable local broadcast). Note that the use of a common transmission rate r bits/sec and a common message size l bits ensures that all messages occupy a certain minimum time $T_l \geq \frac{l}{r}$ on the channel. This extends to messages sent by faulty nodes, because non-faulty nodes can choose to ignore messages that do not conform to the rate/size specification (information about the transmission-rate of the message can be obtained from the recipient's physical layer), giving faulty nodes no incentive to deviate from this established behavior.

The maximum and minimum propagation delays are $d_{prop}{}^{max}$ and $d_{prop}{}^{min}$ respectively (note that $d_{prop}{}^{min} > 0$). Any additional delays in physical layer timestamping are upper-bounded by t_{delay}, yielding a maximum delay bound of $T_d = d_{prop}{}^{max} + t_{delay}$. Thus $T_u = T_l + T_d$.

For the rest of our discussion, we assume that nodes are externally synchronized within bound D. Under this assumption, we may leverage Observation 1.

---

[1]Even in a multi-rate wireless network, it is possible to stipulate as part of the protocol specification that all nodes use a specific transmission rate (say the lowest available) for critical message types that require reliable dissemination.

We seek to ensure that the conditions of Observation 1 from Section 10.3 are satisfied. Thus, we want $2T_l - T_u = T_l - T_d > D$, or $T_l > D + T_d$. Since T_d is independent of T_l, this is always achievable[2] (albeit at the expense of inefficient bandwidth usage) by padding all messages with extra bits to achieve the desired packetsize l (and hence T_l) for the specified transmission rate r. Thus the Receipt-Order Condition can be made to hold.

We now provide a brief description of the message representation.

In order to distinguish between different messages, *distinct* messages sent by a particular source (originator) are distinguished via *identifiers*, that we shall denote as *id*. The *id* is a number in some range $[0, MAX]$, where MAX is a suitably large number. Individual nodes choose the sequence of *id*s for their messages in some *privately determined* pseudo-random manner (such that *id*s are re-used only after large intervals of time; thus identifiers may be considered unique for all practical purposes). This ensures that other nodes have no easy way of anticipating what the sequence of *id*'s for a given source node will be.

If a node sends two conflicting versions of the same message, it implies that they both have the same *id*, but different values. Original messages are represented as $m(src, (id, value))$. Of these, the *src* field is obtained from the MAC header, and thus contains the true MAC address of the node that put the packet on air, since by assumption MAC addresses are not subject to spoofing. The $(id, value)$ part is message-content. If a message m is relayed (repeated) by a neighbor, it is represented as $REPEAT(relay\_src, (m, timestamp))$. Once again, $relay\_src$ is the MAC address of the relay node, obtained from the MAC header. The $(m, timestamp)$ part is message-content (m denotes the $(src, (id, value))$ information for the message; however as this is now part of message *content*, a faulty relay node can modify the *src* information if it so chooses, though it cannot affect the correctness of the $relay\_src$ field in the MAC header).

10.5   The Algorithm

The goal of the algorithm is to achieve the following agreement condition with probabilistic guarantees:

Definition 6. *(Agreement Condition) If a local broadcast source s sends a message, then all*

---

[2]Even if there is some dependence between T_l and T_d, it may still be possible to do so, e.g., if $T_d \leq \alpha T_l + \beta$ where $0 \leq \alpha < 1$ and $\beta \geq 0$ are constants, then one can make the message long enough so that $T_l \geq \frac{D+\beta}{(1-\alpha)}$, and satisy the condition.

its non-faulty neighbors should agree on a single value for this message. If s is non-faulty,
this agreed-upon value should be the one actually sent by s. If s is faulty and sends multiple
conflicting versions of the message, nodes should choose the first *value that s sent.*

For the sake of simplicity and w.l.o.g., we assume that the message m may take one of two values 0 or 1. The algorithm can be easily generalized to more than two message values.

Suppose we have sender s. Each other node u follows the following algorithm:

- On receipt of a message $m(s, (i, p))$ from s directly with (local) receive-timestamp t:

  If no other earlier version of this message (i.e., of the form $m(s, (i, q))$) was received *directly* from s, make note of p as a candidate message value, and re-broadcast a copy of m as $REPEAT(u, (m(s, i, p), t))$. If an earlier version of the same message was received *directly* from s, discard this message.

- On receipt of a message $REPEAT(v, (m(s, i, p), t_v))$:

  If no previous $REPEAT(v, m(s, i, *), *)$ [3] has been received, make note of p as a candidate for message-id i from s, reported by v with timestamp t_v. Keep track of all such copies of m received via $REPEAT$ messages from different repeaters along with their reported timestamps.

  If this was the first message having the form $REPEAT(*, m(s, i, *), *)$ received by the node, start a timer (tagged by (s, i)) to expire after a duration $T + T_u$ (where $T = T_a + T_r$, T_a being the pre-defined access timeout, and T_r being an estimated upper bound on processing time from receiving a message m to time of generating a $REPEAT$ and enqueueing it in the outgoing packet queue).

- On expiration of the timer for (s, i):

  Perform the following filtration and majority-determination procedure on the received $REPEAT$ messages containing repeated messages of the form $m(s, (i, *))$:

  Timestamp-based filtration and majority determination: Let us refer to the value with highest repeated copy count as c_1, and the other one as c_2. If the number of copies of c_2 is less than or equal to b, choose c_1 as the correct value. If the number of copies of c_2 is greater than b: discard any messages with value c_1 whose timestamp t is greater

---

[3] $*$ is a placeholder for any value.

than the timestamps of more than b copies of c_2. Commit to the majority value from amongst the remaining copies of c_1 and c_2.

Theorem 30. *Consider a* local broadcast domain *in the wireless network comprising $nbd(s)$ for some node s. Assume that the physical clocks of all non-faulty nodes satisfy the Receipt-Order Condition. Let α be a constant satisfying $\alpha \leq p_a p_s^2 - \epsilon$, where $\epsilon > 0$ is a constant. If at most b nodes in any single neighborhood are faulty (where $b \leq \left(\frac{\alpha}{1+\alpha}\right) d_o$), then the above algorithm ensures that all non-faulty neighbors of s shall be able to achieve the previously described agreement condition for s's message with error probability at most $d \exp(-\frac{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 d_o}{2(1+\alpha)})$, which is small if d_o is large, and $d_o >> \ln d$.*

Proof. There are two cases: s is non-faulty or s is faulty:

1. *s is non-faulty:* s transmits exactly one version of the message (call it $m_1 = m(s, (i, q_{m_1}))$). Since any $u \in nbd'(s)$ has at most b faulty nodes in $nbd(u)$, it may receive up to a maximum of b spurious repeats of s's message. If the number of $REPEAT$ copies of the message received from non-faulty nodes (and thus containing the correct value) is greater than b, this suffices to distinguish the legitimate value from a spurious one.

2. *s is faulty:* If s is faulty, it may leverage the unreliability of the channel, and attempt to create confusion by sending more than one version of the message, each containing different values. We show that despite this, under the assumed conditions, reliable broadcast will still be achieved.

By assumption, the physical clocks of all non-faulty nodes satisfy the Receipt-Order Condition. Then, in the algorithm described earlier, copies of the second message received from non-faulty neighbors get filtered out as follows:

Suppose the sender s sends the two message-versions $m_1 = m(s, (i, q_{m_1}))$ and $m_2 = m(s, (i, q_{m_2}))$ at absolute times t_1 and t_2 respectively.

Hence, any non-faulty node that receives the first message observes a receive-timestamp that is less than the receive-timestamp for the second message observed by those non-faulty nodes that receive the second message. All non-faulty nodes attach the correct observed timestamp to any $REPEAT$ messages they send, and non-faulty nodes that receive the $REPEAT$ messages record the timestamp along with the message encapsulated in the $REPEAT$.

Recall that the first message-version sent out by s is m_1 and the second is m_2. Also, the message-version with highest pre-filtration count is referred to as c_1 and the other one is referred to as c_2.

We show that if more than b *REPEAT* copies of m_1 were received from non-faulty nodes, the agreement condition is achieved.

Suppose more than b copies of m_1 were received from non-faulty nodes, i.e., more than b *correct* copies of m_1 were received.

Then the following cases may arise:

- If $c_1 = m_1$, and at most b copies of m_2 were received:

  m_1 will win the majority vote, and get chosen immediately.

- If $c_1 = m_1$, i.e., m_1 has the highest pre-filtration count, and greater than b copies of m_2 were received:

  A non-faulty node will only send a *REPEAT* of m_2 if it receives the message m_2 directly from s, and it will affix a correct receive-timestamp to its *REPEAT*. Since the Receipt-Order Condition holds, the timestamp reported in any such *REPEAT* copy of m_2 will be greater than the timestamp reported in any of the correct *REPEAT* copies of m_1. Thus, no more than b copies of $c_2 = m_2$ can bear a false earlier timestamp. Resultantly, no copy of m_1 sent by a non-faulty node will get filtered out erroneously, and m_1 will win the majority vote.

- If $c_1 = m_2$ i.e. m_2 has the highest pre-filtration count:

  Since greater than b copies of m_1 were received from non-faulty nodes, then from the Receipt-Order Condition, any copy (*REPEAT*) of m_2 sent by a non-faulty node has a reported timestamp greater than the reported timestamps on the greater-than-b correct copies of m_1, and the timestamp filtration rule ensures that all copies of m_2 sent by non-faulty nodes get filtered out. This leaves only up to b copies of m_2 sent by faulty nodes. Thus, when the correct *REPEAT* copies of m_1 are greater than b, m_1 will win the majority vote.

Hence, the algorithm definitely makes the correct decision if more than b copies of m_1 were received from non-faulty nodes. This is the same as the sufficient condition we earlier stated for correct decision with a non-faulty source.

When b or fewer copies of m_1 are received from non-faulty nodes, the decision may be correct or wrong, depending on how many copies of m_2 were received.

To bound the error probability, we assume the worst, i.e., it is always wrong if b or fewer copies of m_1 are received from non-faulty nodes.

We represent the copies of m_1 repeated by non-faulty nodes that were received by a node u as a random variable Z. Then, the requirement is that $Z > b$ for both the cases (recall that in the first case, the source is non-faulty, and so it sends only one message-version m_1, but up to b spurious $REPEAT$ messages containing wrong values may still be received from faulty nodes).

Let the number of non-faulty mutual neighbors of s and u be g. Then $g \geq d_o - b$. Z is the sum of g i.i.d. $Bernoulli(p_a p_s^2)$ random variables, since a repeated copy of m_1 is received from a non-faulty neighbor if that neighbor received m_1 directly from s (probability p_s), it was able to transmit the $REPEAT$ packet before timeout (probability p_a), and the $REPEAT$ was successfully received by u (probability p_s). This allows us to apply the following special form of the Chernoff bound [83]:

$$\Pr[Z \leq (1-\beta)E[Z]] \leq \exp(\frac{-\beta^2 E[Z]}{2}), 0 < \beta < 1 \tag{10.1}$$

Knowing that $b \leq \frac{\alpha}{1+\alpha} d_o \leq \alpha g$, we can set $\beta = 1 - \frac{\alpha}{p_a p_s^2}$ to obtain $b \leq (1-\beta)E[Z]$. Thus application of the Chernoff bound[4] yields:

$$
\begin{aligned}
\Pr[Z \leq b] &\leq Pr[Z \leq (1-\beta)E[Z]] \\
&\leq \exp(-\frac{(1 - \frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 g}{2}) \\
&\leq \exp(-\frac{(1 - \frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 d_o}{2(1+\alpha)})
\end{aligned}
\tag{10.2}
$$

Applying the union bound over all d neighbors of sender s, probability that any node makes an error is at most $d \exp(-\frac{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 d_o}{2(1+\alpha)})$, which is small for large d_o, and $d_o >> \ln d$. □

Note that, as d increases, the timeout component T_a would typically also need to increase to maintain a sufficiently high value of p_a (due to increased contention for the shared

---

[4]Since we need $\beta > 0$ for application of the Chernoff Bound, this yields the constraint that $\alpha \leq p_a p_s^2 - \epsilon$ with $\epsilon > 0$. Thus α (which gives a measure of the proportion of tolerable faults) can be large when the probability of successful receipt ($p_a p_s^2$) is large, and can only be small when $p_a p_s^2$ is small. Also note that these constants determine how much larger d_o should be compared to d to achieve small error probability.

channel). However, in most cases of practical interest, d will not be unduly large, and a moderate value for T can suffice. Besides, the protocol is still fairly scalable, as it only requires one message to be sent by each node.

In our analysis, we have assumed that whenever the number of copies of m_1 received from non-faulty nodes is less than b, a wrong decision is made. In actuality, if the number of copies of m_1 received from non-faulty nodes is less than b, there may still be situations where a correct decision may be made (it is possible that the total number of received copies of m_2 (from faulty or non-faulty nodes) may be much less than b, since these transmissions are also subject to errors in reception). Thus, the presented analysis establishes a conservative upper bound on the error probability.

10.6   Possible Optimizations

From a practical perspective, one can consider many possible enhancements/optimizations to the basic algorithm.

1. Each node can be made to retransmit its $REPEAT$ messages k times. This can help improve loss-resilience, without causing duplication problems, since, in the absence of address spoofing (which is one of our assumptions), two receipts of the same message are easily identified by the repeater's address, and extra copies discarded.

2. One could consider triggering the reliable local broadcast algorithm only if at least one warning message is heard from a node claiming to have heard two inconsistent messages sent by s (this would work only if it is very likely that a fair number of nodes will receive both variants of s's message). Also, while faulty nodes can raise false alarms, that is no worse that proactively using the algorithm each time.

10.7   Discussion on Synchronization Requirements

The synchronization assumptions required to ensure the Receipt-Order Condition holds may actually be practically feasible in many settings.

One can envision future scenarios where wireless nodes may be equipped with on-chip atomic clocks [56] with very low drift. Thus, if the clocks are synchronized with an external time source at time of deployment, then one might bound the total skew over the entire

operational lifetime of the network, and this would not be overly large. Alternatively, nodes might be GPS-equipped, thus providing an out-of-band means of external synchronization. In such scenarios, the conditions for Observation 1 can be made to hold.

In the absence of on-chip atomic clocks or GPS-equipped devices, it may not be possible to ensure that all nodes in the network be synchronized to an external clock within some constant bound D. However, it may still be quite feasible to ensure that each node is internally synchronized within constant bound D with its two-hop neighbors. One could envisage a situation where nodes are initially synchronized at time of deployment, and thereafter periodically run a re-synchronization protocol, to ensure that any any two nodes within two-hops of each other always stay internally synchronized within the bound D.

A lightweight Byzantine time synchronization protocol might possibly suffice for this. In the period between two consecutive re-synchronizations, the conditions of Observation 2 can thus be made to hold for every local broadcast domain in the network.

10.8 Using the Primitive for Multi-Hop Broadcast

We briefly discuss how the proposed primitive could potentially be used as a building block in a protocol to achieve broadcast in a multi-hop setting. As was mentioned earlier, the algorithm we described in Section 8.4 was used as a subroutine in the bounded-collision-resilient algorithm of [58]. It was observed in [58] that this algorithm requires neighbors of the original sender to agree on the value it sent, even if the original sender is faulty; for other nodes in the network, correctness of the algorithm only requires that neighbors of non-faulty nodes agree on the messages they (the non-faulty nodes) send, and this property was exploited. It follows that, if one is using a global broadcast protocol with similar properties, one could consider using the reliable local broadcast primitive in the neighborhood of the original sender, and merely stipulate that other nodes retransmit their messages a sufficient number of times.

Otherwise, if the protocol requires that neighbors of all nodes agree on what they sent, one could potentially proceed as follows: Let us consider a multi-hop network of n nodes, where the minimum node degree is d_{min}, maximum node degree is d_{max}, and $d_o = \min_{x} \min_{y \in nbd'(x)} |nbd'(x) \cap nbd'(y)|$. Thus d_o is the minimum number of common neighbors shared by any two neighbors. The number of faulty nodes in any single neighborhood is at

most $b \leq \frac{\alpha}{1+\alpha} d_o$ where $\alpha \leq p_a p_s^2 - \epsilon (\epsilon > 0)$. Through exchange of periodic hello messages, nodes maintain a list of neighbors. Neighbors are added/removed only if more than a certain number of HELLO messages have been consecutively received/lost. This helps maintain a degree of stability in the neighborhood information, in the face of short-term signal fluctuations. Suppose we have a global multi-hop broadcast protocol that assumes reliable local broadcast, and requires a total of $O(n^m)$ messages to be sent (m is a constant), i.e. has message complexity polynomial in n. Then, for each step of the protocol that requires a node to perform a local broadcast, the reliable local broadcast primitive protocol is run in the *local broadcast domain* comprising the node and its neighbors. Following the proof argument of Theorem 30, we can obtain that the probability local broadcast is achieved reliably is at least $1 - d_{max} \exp(-\frac{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 d_o}{2(1+\alpha)}) = 1 - \exp(-\frac{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 d_o}{2(1+\alpha)} + \ln d_{max})$. Since n^m such successful local broadcasts are needed, if $d_o = c_1 m \log n$ for a suitably chosen constant $c_1 > \frac{2(1+\alpha)}{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2}$, and $d_{max} \leq c_2 \log n$ for another suitably chosen constant c_2 (note that $c_2 \geq c_1 m$ by definition), then by applying the union bound, one may see that the global broadcast will also succeed with probability at least $1 - n^m \exp(-\frac{(1-\frac{\alpha}{p_a p_s^2})^2 p_a p_s^2 d_o}{2(1+\alpha)} + \ln d_{max})$, which approaches 1 for large n.

The tolerable number of per-neighborhood faults would be given by the minimum of the tolerance threshold for the global protocol, and the local broadcast primitive.

10.9   Discussion

The algorithm we have outlined in this chapter is primarily an exploratory proof-of-concept approach, whereby we have sought to highlight the potential for leveraging the shared nature of the medium in conjunction with knowledge of physical layer characteristics (in this case, the transmission rate), and other information from lower-layers (in this case, timestamps), to achieve useful message-ordering conditions, which can facilitate the design of scalable probabilistic solutions to the reliable local broadcast problem, and possibly other reliable communication primitives. However, there are still numerous outstanding issues that need to be addressed.

One issue is that of using a suitable Byzantine time synchronization protocol to ensure internal synchronization between neighboring nodes (see Section 10.7). It might be possible to leverage existing work in this area, e.g., [107]. Another issue is that one might wish

to eliminate the requirement in Observation 2 that during the interval in which the local broadcast is occurring, nodes do not adjust their clocks. This would require a synchronization algorithm that can run simultaneously with the local broadcast algorithm without affecting the Receipt-Order Condition. Additionally, the described algorithm assumes i.i.d. loss probabilities. If channel losses exhibit spatial correlation, the algorithm may need to be modified to handle such situations.

A major shortcoming of the algorithm is the need to estimate the timeout T based on access probability p_a, average length of outgoing packet-queues, and processing time to generate a *REPEAT*. It would be preferable to have an algorithm where nodes decide to invoke the filtration and majority determination procedure based on some event, e.g., receipt of certain messages.

Many of the assumptions in this chapter are justified by assuming a network with a single channel and omnidirectional antennas. Also relevant are alternative scenarios where multiple channels or beam-forming antennas are available. We remark that usage of multiple channels or directional antennas tends to alter the broadcast nature of the wireless medium, and makes the network look increasingly like a point-to-point network. Thus, algorithms based on the point-to-point abstraction may increasingly seem suitable in such scenarios.

Furthermore, as mentioned in Section 7.3, the issue of handling a bounded number of collisions in a grid network when the channel is reliable was addressed in [58]. It is relevant to consider the possibility of combining ideas from [58] with some of the ideas discussed in this chapter, to handle both an unreliable channel and a bounded number of collisions. Other possibilities include trying to exploit the availability of multiple channels (as in [27]), or other forms of physical layer diversity.

Chapter 11

Conclusion

In this dissertation we have investigated the performance of wireless networks that are subject to miscellaneous forms of functional constraints or malfunction. As wireless networks proliferate and find use in diverse scenarios, they will increasingly need to operate in the presence of heterogeneous (and often constrained) hardware capabilities. Furthermore, fault-tolerant communication algorithms will be required to provide the building blocks for reliable operation in the face of failure and/or disruption. The research performed as part of this dissertation has contributed to developing an understanding of some of the issues that would arise in such scenarios.

We have examined the routing and scheduling implications of having heterogeneous radios with constrained switching ability, and channels with heterogeneous characteristics, through theoretical investigation. The asymptotic capacity results in Chapter 3 and Chapter 4 quantify the impact of channel switching constraints, and also provide intuition about the implications of such switching constraints for load-balanced routing and scheduling. The results in Chapter 5 provide insight regarding suitable packet scheduling strategies for networks where channels can have diverse rate characteristics.

The channel and interface management protocol described in Chapter 6 provides a proof-of-concept of the possibility of evolving a generalized conceptual design approach toward handling various kinds of physical layer heterogeneity.

The broadcast results in Chapter 8 and Chapter 9 establish fundamental limits on fault-tolerance and also provide insight into the potential for exploiting the broadcast nature of the wireless medium for reliable communication.

Some of the theoretical results that are part of this dissertation have also served as building blocks for other work. The asymptotic capacity results for random (c, f) assignment that were described in Chapter 4 have been used to obtain asymptotic capacity results

with random key pre-distribution in [11]. The algorithm for broadcast with locally-bounded faults is used in [58] as a subroutine in a broadcast algorithm that is resilient to an adversary that can cause a bounded number of collisions.

We have also identified and discussed many interesting directions for future work building upon this research, both in terms of theory and protocol design.

Appendix A

Notation and Terminology

Throughout the text of this dissertation, we have used the following standard asymptotic notation:

- $f(n) = O(g(n))$ means that $\exists\ c > 0, N_o > 0$, such that $f(n) \leq cg(n)$ for all $n \geq N_o$

- $f(n) = o(g(n))$ means that $\lim\limits_{n \to \infty} \frac{f(n)}{g(n)} = 0$

- $f(n) = \omega(g(n))$ means that $g(n) = o(f(n))$

- $f(n) = \Omega(g(n))$ means that $g(n) = O(f(n))$

- $f(n) = \Theta(g(n))$ means that $\exists\ c_1 > 0, c_2 > 0, N_o > 0$, such that $c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq N_o$

When $f(n) = O(g(n))$, any function $h(n) = O(f(n))$ is also $O(g(n))$. We often refer to such a situation as $h(n) = O(f(n)) \implies O(g(n))$.

Whenever we use the notation "log" without explicitly specifying the base, we imply the natural logarithm. We also use the notation "ln" for the natural logarithm in many proofs. We explicitly specify the base whenever it is other than e (the base of the natural logarithm).

When we use the term *w.h.p.* (with high probability), we imply *with probability that tends to 1 as n tends to ∞* (where n is as defined in the specific context).

Appendix B

Proofs of Connectivity Results

The necessary conditions for connectivity with adjacent (c, f) assignment and random (c, f) assignment are both obtained by an adaptation of the proof techniques used in [42] to obtain the necessary condition for connectivity. The major difference stems from the fact that in the presence of switching constraints, two nodes may be within range and yet be unable to communicate with each other (if they cannot switch(operate) on any common channel).

The following lemma which was stated and proved in [42] will be used in our proofs.

Lemma 48. *(i) For any $p \in [0, 1]$*

$$(1 - p) \leq e^{-p}$$

(ii) For any given $\theta \geq 1$, there exists $p_0 \in [0, 1]$, such that

$$e^{-\theta p} \leq (1 - p), \quad \forall 0 \leq p \leq p_0$$

If $\theta > 1$, then $p_0 > 0$.

Proof. See Lemma 2.1 in [42]. $\square$

Lemma 49. *If $\pi r^2(n) = \frac{(\log n + b)}{pn}$, then, for any fixed $\theta < 1$:*

$$n(1 - p\pi r^2(n))^{(n-1)} \geq \theta e^{-b} \tag{B.1}$$

for sufficiently large n.

Proof. This is basically the proof of Lemma 2.2 from [42], as presented in [42], with the minor change that $\pi r^2(n)$ is replaced with $p\pi r^2(n)$. Taking the log of the L.H.S. and using

the Taylor Series expansion, we have:

$$\log L.H.S. = \log n + (n-1) \log \left(1 - p\pi r^2(n)\right)$$

$$= \log n - (n-1) \sum_{i=1}^{\infty} \frac{(p\pi r^2(n))^i}{i}$$

$$= \log n - (n-1) \left(\sum_{i=1}^{2} \frac{(\log n + b)^i}{in^i} + \epsilon(n) \right)$$

$$\text{where } \epsilon(n) = \sum_{i=3}^{\infty} \frac{p\pi r^2(n))^i}{i} = \sum_{i=3}^{\infty} \frac{(\log n + b)^i}{in^i}$$

$$\leq \frac{1}{3} \int_{i=2}^{\infty} \left(\frac{\log n + b}{n} \right)^x dx$$

$$\leq \frac{1}{3} \left(\frac{\log n + b}{n} \right)^2$$

for large n

From the above, we obtain:

$$\log L.H.S. \geq \log n - (n-1) \left(\frac{\log n + b}{n} + \frac{5(\log n + b)^2}{6n^2} \right)$$

$$\geq -b - \frac{(\log n + b)^2 - (\log n + b)}{n} \geq -b - \delta$$

Setting $\delta = \ln \frac{1}{\theta}$, and taking exponents on both sides yields that the L.H.S. $\geq \theta e^{-b}$ for large n. $\qquad\square$

B.1   Adjacent (c, f) Assignment: Proof of Theorem 1

Given that a node has block location i, the probability that it can operate on a common channel with another node (we shall often refer to this as sharing a channel) within its range is given in (3.3), and denoted by $p_{adj}(i)$.

Note that $p_{adj}(i)$ is different for different block locations i primarily because nodes with channel-blocks at the fringes of the band are less likely to share channels with other nodes. Since we are deriving a *necessary* condition for connectivity, it is possible to make the following assumption for the purpose of this proof:

Channel pairs $(i, c - f + i + 1), 1 \leq i \leq f - 1$ possess magical capabilities, such that

communication on channel i ends up being visible on channel $c - f + i + 1$, and vice-versa. Thus, if a node has channel i, then it can also communicate with a node that does not share any channel with it, but has channel $c - f + i + 1$. Another way to view this situation is that although nodes are assigned channels as per the adjacent (c, f) model, at time of network operation, a node having channel $c - f + i + 1, 1 \leq i \leq f - 1$ uses channel i instead (i.e., $c - f + i + 1$ serves as an alias for i).

Under this assumption, $p_{adj}(i) = \min\{\frac{2f-1}{c-f+1}, 1\}$, for all i. If the network is disconnected under this assumption, then it must necessarily be so otherwise. This can be seen thus: suppose we are given a network instance with nodes assigned adjacent channels as per the adjacent (c, f) model, and we then impose the assumption stated above. Suppose this network is disconnected. Now the imposed assumption is removed, but the channel block assigned to each node remains unchanged. Then, in the new scenario, some nodes that were earlier able to communicate, will not be able to do so anymore; however those nodes that were incapable of communicating will preserve their status quo. Thus, a necessary condition for the hypothetical network is also valid for the actual network.

Therefore, to establish a necessary condition for connectivity with adjacent (c, f) assignment, we estabslish a necessary condition for connectivity in a scenario where we have the additional assumption described above. This proof is an adaptation of a similar proof in [42] (Theorem 2.1 in [42].

We focus on the disconnection event where singleton sets are partitioned from the rest of network. Recall that $p = \min\{\frac{2f-1}{c-f+1}, 1\}$. When $f \geq \frac{c+2}{3}$, then $p = 1$, i.e., any pair of nodes that are within range can communicate with each other, and the necessary condition result from [42] applies directly. Hence, we consider only the scenario $f < \frac{c+2}{3}$ for which $p = \frac{2f-1}{c-f+1}$. Also note that:

$$\pi r^2(n) \leq \frac{2 \log n}{pn} \leq \frac{2\alpha \log^2(n)}{n} \quad \text{where } \alpha \text{ is a constant}$$

$$(\because p \geq \frac{1}{c - f + 1} > \frac{1}{c} \text{ and } c \leq \alpha \log n \text{ for some constant } \alpha \quad \text{(B.2)}$$

$$\text{and } b(n) < \log n \text{ for large } n \quad \because \limsup_{n \to \infty} b(n) < +\infty)$$

The probability that a node x is isolated, i.e., cannot communicate with any other node, is given by $p_1 = (1 - p\pi r^2(n))^{(n-1)}$. Consider the event that nodes x and y are both isolated.
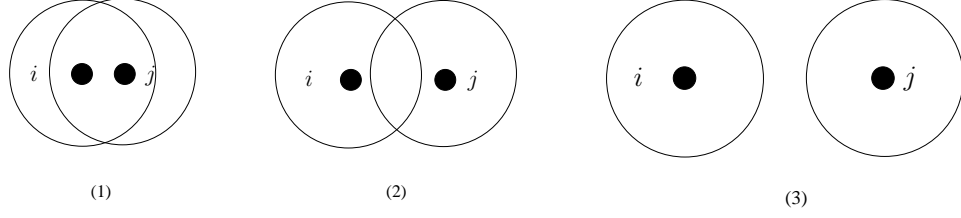
251

Figure B.1: Three Cases: Necessary Condition for Connectivity

There are three different cases for this (also see Fig. B.1):

1. x and y lie within distance $r(n)$ of each other, but do not share a common channel

2. x and y do not lie within distance $r(n)$ of each other, but have overlapping neighborhood regions, i.e., they lie within a distance $2r(n)$ of each other

3. The neighborhood regions of x and y are disjoint, i.e., the distance between x and y is greater than $2r(n)$.

The probability that both x and y are isolated is given by the probability that they cannot communicate with each other, and none of the remaining $n-2$ nodes can communicate with either of them.

From the geometry of the situation (Fig. B.2), it follows that if x and y are separated by a distance $d(n)$ then the overlap area between the neighborhoods of x and $y = 2\,[(\text{area of quadrant subtending angle } 2\theta) - (\text{ area of } \triangle ABC)] = 2r^2(n)\theta - r^2(n)sin(2\theta)$, where $\theta = \cos^{-1}\left(\frac{d(n)}{2r(n)}\right)$.

Let us first consider case 1, i.e., the distance between x and y is $d(n) \leq r(n)$. We view it as two sub-cases (noting that $\frac{16\log\log n}{\log n} < 1$ for large n):

- (i) y is at distance $d(n) \leq r'(n) = \left(\frac{16\log\log n}{\log n}\right)r(n)$ of x

- (ii) y is at distance $d(n) > r'(n) = \left(\frac{16\log\log n}{\log n}\right)r(n)$ of x

The probability that a node $z \neq x, y$ within range of both x and y is capable of communicating with at least one of x and y, given that x, y cannot communicate with each other is $q \geq \frac{\min\{3f-1, c-f+1\}}{c-f+1}$. Also, when $f \leq \frac{c+2}{4}$, then $3f - 1 \leq c - f + 1$, and $q \geq \frac{3f-1}{c-f+1} \geq \frac{3p}{2}$.

For sub-case (i) of case (1), the overlap area between the neighborhoods of x and y is at least $(1 - \delta)\pi r^2(n)$ for any $\delta > 0$ and large enough n, since the separation $d(n) \leq \left(\frac{16\log\log n}{\log n}\right)r(n)$. For our purpose, it suffices to take $\delta = \frac{1}{5}$, yielding an overlap area of at
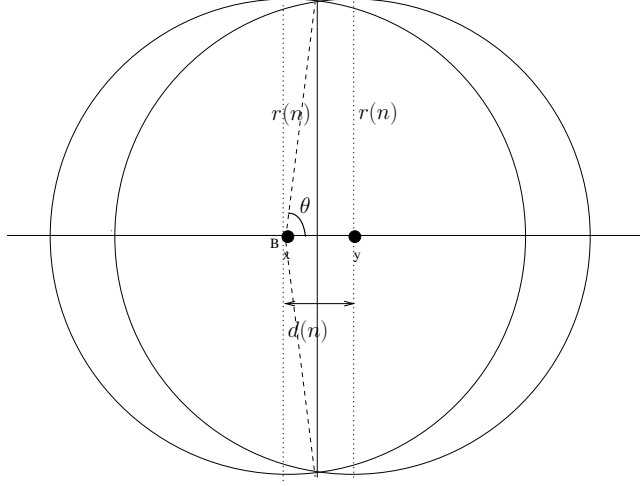
252

Figure B.2: Overlap Area of Neighborhoods

least $\frac{4\pi r^2(n)}{5}$. Then the probability that a node can communicate with either x or y or both is at least q times the probability of lying in the overlap area.

Thus, the contribution of subcase (i) of case (1) to the probability that both x and y are isolated can be upper-bounded as follows:

When $f \leq \frac{c+2}{4}$ (implying $q \geq \frac{3p}{2}$):

$$
\begin{aligned}
p_{21(i)} &\leq \pi r'^2(n)(1-p)\left(1 - q\frac{4\pi r^2(n)}{5}\right)^{n-2} \\
&< \pi r^2(n)\left(1 - \frac{4q\pi r^2(n)}{5}\right)^{n-2} \\
&\leq \pi r^2(n)\left(1 - \frac{6p\pi r^2(n)}{5}\right)^{n-2} \\
&\leq \pi r^2(n)e^{-(n-2)\frac{6}{5}p\pi r^2(n)} \qquad \text{from Lemma 48} \\
&\leq \frac{2\alpha \log^2 n}{n}e^{-(n-2)\frac{6(logn+b(n))}{5n}} \qquad \text{from (B.2)} \\
&= e^{-\frac{6(logn+b(n))}{5} + \frac{12(\log n+b(n))}{5n} + \log 2\alpha + 2\log\log n - \log n} \\
&= e^{-\frac{11\log n}{5} - \frac{6b(n)}{5} + \frac{12(\log n+b(n))}{5n} + \log 2\alpha + 2\log\log n} \\
&\leq e^{-\frac{21\log n}{10} - b(n)} \qquad \text{for large } n \\
&\leq e^{-2\log n - b(n) - \frac{1}{2}\log\log n} \qquad \text{for large } n
\end{aligned}
$$

(B.3)

When $f > \frac{c+2}{4}$, $p = \min\{\frac{2f-1}{c-f+1}, 1\} \geq \frac{1}{2}, \forall c \geq 2$. For this situation, we merely consider the probability that one of the remaining $n-2$ nodes can communicate with one of x and y
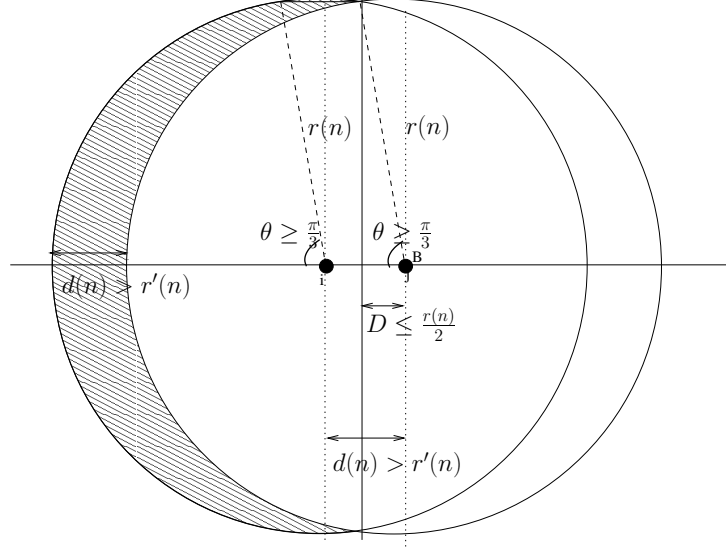
Figure B.3: First Case: Necessary Condition for Connectivity

(say x) to obtain the upper bound on both x and y being isolated:

$$p_{21(i)} \leq \pi r'^2(n)(1-p)(1-p\pi r^2(n))^{n-2}$$

$$\leq \left(\frac{256(\log\log n)^2}{\log^2 n}\right)\pi r^2(n)(1-p\pi r^2(n))^{n-2}$$

$$\leq \left(\frac{256(\log\log n)^2}{\log^2 n}\right)\pi r^2(n)e^{-(n-2)p\pi r^2(n)} \qquad \text{from Lemma 48}$$

$$\leq \left(\frac{256(\log\log n)^2}{\log^2 n}\right)\left(\frac{\log n + b(n)}{pn}\right)e^{-(n-2)\frac{(logn+b(n))}{n}} \qquad (B.4)$$

$$\leq \left(\frac{256(\log\log n)^2(2(2\log n))}{n\log^2(n)}\right)e^{-(n-2)\frac{(logn+b(n))}{n}} \qquad \because p \geq \frac{1}{2}$$

$$\leq e^{-\log n - b(n) + \frac{2(logn+b(n))}{n} + \log 256 + \log 4 - \log n - \log\log n + 2\log\log\log n}$$

$$\leq e^{-2\log n - b(n) - \frac{1}{2}\log\log n} \text{ for large } n$$

From B.3 and B.4, for all valid f:

$$p_{21(i)} \leq e^{-2\log n - b(n) - \frac{1}{2}\log\log n} \text{ for large } n \qquad (B.5)$$

For sub-case (ii), the situation is depicted in Fig. B.3. The probability that some node can communicate with at least one of x or y is lower bounded by the probability that it lies in range of x (this probability is $\pi r^2(n)$) and shares a channel with it (this probability is p), or it lies out of range of x but within range of y (this probability is at least $\frac{\sqrt{3}r(n)r'(n)}{2}$ for

large enough n)[1], and shares a channel with y (this probability is p). The contribution to the probability that both x and y are isolated is thus at most:

$$p_{21(ii)} \leq \left(\pi r^2(n) - \pi r'^2(n)\right)(1-p)\left(1 - p\left(\pi r^2(n) + \frac{\sqrt{3}r(n)r'(n)}{2}\right)\right)^{n-2}$$

$$\leq \pi r^2(n)\left(1 - p\left(\pi r^2(n) + \frac{\sqrt{3}r(n)r'(n)}{2}\right)\right)^{n-2}$$

$$\leq \pi r^2(n)\left(1 - p\pi r^2(n)\left(1 + \frac{\sqrt{3}r'(n)}{2\pi r(n)}\right)\right)^{n-2}$$

$$\leq \pi r^2(n)\left(1 - p\pi r^2(n)\left(1 + \frac{8\sqrt{3}\log\log n}{\pi\log n}\right)\right)^{n-2} \tag{B.6}$$

$$\leq \pi r^2(n)e^{-(n-2)p\pi r^2(n)(1+\frac{4\log\log n}{\log n})} \qquad \text{from Lemma 48} \quad (\because \pi < 2\sqrt{3}))$$

$$\leq \left(\frac{2\alpha\log^2 n}{n}\right)e^{-(n-2)p\pi r^2(n)(1+\frac{4\log\log n}{\log n})} \qquad \text{from (B.2)}$$

$$\leq e^{-(n-2)p\pi r^2(n)(1+\frac{4\log\log n}{\log n})+\log 2\alpha+2\log\log n - \log n}$$

$$\leq e^{-\log n - b(n) - 4\log\log n + \frac{2(\log n + b(n))(1+\frac{4\log\log n}{\log n})}{n} + \log 2\alpha + 2\log\log n - \log n}$$

$$\leq e^{-2\log n - b(n) - \log\log n} \text{ for large } n$$

For case 2, the probability that some node can communicate with either x or y can be lower bounded by the probability that it lies in range of x (this probability is $\pi r^2(n)$) and shares a channel with it (this probability is p), or it lies out of range of x but within range of y (the disjunction of the two circles in Fig. B.2 is at least $\frac{1}{2}\pi r^2(n)$ for this case), and shares a channel with it. Thus the contribution of this case to the probability that both x and y are isolated is upper bounded by:

$$p_{22} = (4\pi r^2(n) - \pi r^2(n))(1 - \frac{3}{2}p\pi r^2(n))^{n-2}$$

$$\leq 3\pi r^2(n)e^{-\frac{3(n-2)p\pi r^2(n)}{2}} \qquad \text{from Lemma 48}$$

$$\leq \left(\frac{6\alpha\log^2 n}{n}\right)e^{-(n-2)\frac{3(\log n + b(n))}{2n}} \text{ from (B.2)}$$

$$\leq e^{-\frac{3}{2}\log n - \frac{3}{2}b(n) + \frac{3(\log n + b(n))}{n} + \log 6\alpha + 2\log\log n - \log n}$$

---

[1] The area within range of y but out of range of x is given by $\pi r^2(n) -$ overlap area ; where overlap area $= 2$ (area of quadrant subtending angle $2\theta -$ area of $\triangle ABC$) $\leq \pi r^2(n) - r^2(n)\sin(2\theta)$. Note that $\frac{\pi}{3} \leq \theta \leq \frac{\pi}{2}$. Thus the non-overlap area $\geq r^2(n)\sin(2\theta) = r^2(n)(2\sin\theta\cos\theta) = r^2(n)2\sin\theta\frac{d(n)}{2r(n)} \geq 2r^2(n)\left(\sin\frac{\pi}{3}\right)\frac{r'(n)}{2r(n)} \geq \frac{\sqrt{3}r(n)r'(n)}{2}$

$$\leq e^{-\frac{9}{4}\log n - \frac{3}{2}b(n)} \text{ for large } n \tag{B.7}$$

For case 3, the probability that both x and y are isolated is upper bounded by:

$$
\begin{aligned}
p_{23} &= (1 - 4\pi r^2)(1 - p(2\pi r^2(n)))^{n-2} \\
&\leq (1 - 2p\pi r^2(n))^{n-2} \\
&\leq e^{-2(n-2)p\pi r^2(n)} \text{ from Lemma 48} \\
&\leq e^{-2\log(n) - 2b(n) + \frac{4(\log n + b(n))}{n}}
\end{aligned}
\tag{B.8}
$$

Then, the probability p_2 that nodes x and y are both isolated is given by:

$$p_2 \leq p_{21(i)} + p_{21(ii)} + p_{22} + p_{23} \tag{B.9}$$

Let us first consider the case where $b(n) = b$ is a constant.

$$
\begin{aligned}
\Pr[\text{ disconnection }] &\geq \sum_x \Pr[x \text{ is only isolated node}] \\
&\geq \sum_x \Pr[x \text{ isolated }] - \sum_{x,y\neq x} \Pr[x \text{ and } y \text{ both isolated }] \\
&= np_1 - n(n-1)p_2 \\
&\geq n(1 - p\pi r^2(n))^{(n-1)} - n(n-1)\left(p_{21(i)} + p_{21(ii)} + p_{22} + p_{23}\right) \\
&\geq \theta e^{-b} - n(n-1)\left(e^{-2\log n - b - \frac{1}{2}\log\log n}\right. \\
&\qquad \left. + e^{-2\log n - b - \log\log n} + e^{-\frac{9}{4}\log n - b} + e^{-2\log n - 2b + \frac{4(\log n + b)}{n}}\right) \\
&\geq \theta e^{-b} - (1+\epsilon)e^{-2b} \\
&\quad \text{for any } \theta < 1, \epsilon > 0, \text{ and large } n \text{ (Lemma 48, Lemma 49)}
\end{aligned}
\tag{B.10}
$$

Now consider the case where $b(n)$ is not constant, and $\limsup\limits_{n\to\infty} b(n) = b$. Then, for any $\epsilon > 0$, $b(n) - b \leq \epsilon$ for large n. Since the probability of disconnection monotonically decreases in $b(n)$, we can take the following bound:

$$\Pr[\text{disconnection}] \geq \theta e^{-(b+\epsilon)} - (1+\epsilon)e^{-2(b+\epsilon)} \tag{B.11}$$
$$(\text{ for large enough } n)$$

Since (B.10) and (B.11) hold for all any $\theta < 1, \epsilon > 0$ and large enough n, it follows that when $\limsup\limits_{n\to\infty} b(n) < +\infty$, the network is asymptotically disconnected with some positive probability.

B.2 Random (c, f) Assignment: Proof of Theorem 4

From the model definition, the probability that two nodes in range of each other can operate on a common channel (we will often refer to this as sharing a channel) is $p = p_{rnd}$ where $1 - p_{rnd} = (1 - \frac{f}{c})(1 - \frac{f}{c-1})...(1 - \frac{f}{c-f+1})$. Note that for $f > \frac{c}{2}$, $p = p_{rnd} = 1$, as any two nodes are guaranteed to have at least one common channel. Then the necessary condition for connectivity proved in [42] is applicable. Therefore, we will only consider the case $f \leq \frac{c}{2}$.

The probability that a node x is isolated, i.e., cannot communicate with any other node is give by $p_1 = (1 - p\pi r^2(n))^{(n-1)}$.

We begin by making the following observations:

$$p = p_{rnd} \geq \frac{f}{c} \tag{B.12}$$

$$\pi r^2(n) \leq \frac{2c \log n}{f} \leq \frac{2\alpha \log^2(n)}{n} \text{ for some constant } \alpha$$

$$\because c = O(\log n) \implies c \leq \alpha \log n \text{ for some constant } \alpha \text{ and large enough } n \tag{B.13}$$

$$\text{and } b(n) < \log n \text{ for large } n \quad \because \limsup_{n\to\infty} b(n) = b < +\infty$$

Consider the event that two nodes x and y are both isolated. There are three different cases for this (Fig. B.1):

1. x and y lie within distance $r(n)$ of each other, but do not share a common channel

2. x and y do not lie within distance $r(n)$ of each other, but have overlapping neighborhood regions, i.e. lie within distance $2r(n)$ of each other

3. The neighborhood regions of x and y are disjoint, i.e., the distance between them is greater than $2r(n)$.

From the geometry of the situation (Fig. B.2), it follows that if x and y are separated by a distance $d(n)$ then the overlap area between the neighborhoods of x and $y = 2$ (area

of quadrant subtending angle 2θ − area of $\triangle ABC$) = $2r^2(n)\theta - r^2(n)sin(2\theta) \leq \pi r^2(n) - r^2(n)\sin(2\theta)$, where $\theta = \cos^{-1}\frac{d(n)}{2r(n)}$.

Of these, for case (1), consider two sub-cases:

- (i) y is at distance $d(n) \leq r'(n) = \left(\frac{16 \log \log n}{\log n}\right) r(n)$ from x

- (ii) y is at distance $d(n) > r'(n) = \left(\frac{16 \log \log n}{\log n}\right) r(n)$ from x

The probability that a node $z \neq x, y$ within range of both x and y is capable of communicating with at least one of x and y, given that they do not have a common channel of operation, is given by $q = 1 - (1 - \frac{2f}{c})(1 - \frac{2f}{c-1})...(1 - \frac{2f}{c-f+1}) \geq p$ (recall that we are only considering $f \leq \frac{c}{2}$).

When $f \geq \frac{c-f+1}{2}$, it is evident that $q = 1 \geq p$. When $f < \frac{c-f+1}{2}$:

$$
\frac{1-p}{1-q} = \frac{(1 - \frac{f}{c})(1 - \frac{f}{c-1})...(1 - \frac{f}{c-f+1})}{(1 - \frac{2f}{c})(1 - \frac{2f}{c-1})...(1 - \frac{2f}{c-f+1})}
$$

$$
= \left(1 + \frac{\frac{f}{c}}{1 - \frac{2f}{c}}\right)\left(1 + \frac{\frac{f}{c-1}}{1 - \frac{2f}{c-1}}\right)...\left(1 + \frac{\frac{f}{c-f+1}}{1 - \frac{2f}{c-f+1}}\right)
$$

$$
\geq 1 + \frac{\frac{f}{c}}{1 - \frac{2f}{c}} + \frac{\frac{f}{c-1}}{1 - \frac{2f}{c-1}} + ... + \frac{\frac{f}{c-f+1}}{1 - \frac{2f}{c-f+1}}
$$

$$
\geq 1 + \frac{f}{c} + \frac{f}{c-1} + ... + \frac{f}{c - f + 1}
$$

$$
\geq 1 + \frac{f^2}{c}
$$

(B.14)

Hence:

$$
q \geq 1 - \frac{1-p}{1 + \frac{f^2}{c}} = p + (1-p) - \frac{1-p}{1 + \frac{f^2}{c}}
$$

$$
= p + (1-p)\left(1 - \frac{1}{1 + \frac{f^2}{c}}\right) = p\left(1 + \frac{(\frac{1}{p} - 1)\frac{f^2}{c}}{1 + \frac{f^2}{c}}\right)
$$

$$
= p\left(1 + \frac{\frac{1}{p} - 1}{\frac{c}{f^2} + 1}\right) \geq p\left(1 + \frac{\frac{c}{2f^2} - 1}{\frac{c}{f^2} + 1}\right) \quad \text{from Lemma 14 and the fact that } p = p_{rnd}
$$

$$
\geq p\left(1 + \frac{\frac{c}{2f^2}(1 - \frac{2f^2}{c})}{\frac{c}{f^2}(1 + \frac{f^2}{c})}\right)
$$

(B.15)

For sub-case (i) of case (1), the overlap area between the neighborhoods of x and y is at

258

least $(1 - \delta)\pi r^2(n)$ for any $\delta > 0$ and large enough n, since the separation $d(n) \leq r'(n) = \left(\frac{16 \log \log n}{\log n}\right) r(n)$. For our purpose, it suffices to take $\delta = \frac{1}{16}$, yielding an overlap area of at least $\frac{15\pi r^2(n)}{16}$. Then the probability that a node can communicate with either x or y or both is at least q times the probability of lying in the overlap area.

When $\frac{f}{c} \leq \frac{(\log \log n)^3}{\log n}$, then from (B.15):

$$
\begin{aligned}
q &\geq p\left(1 + \frac{\frac{c}{2f^2}(1 - \frac{2f^2}{c})}{\frac{c}{f^2}(1 + \frac{f^2}{c})}\right) \\
&\geq p\left(1 + \frac{1}{3}\right) = \frac{4p}{3} \quad \text{for } \frac{f}{c} \leq \frac{(\log \log n)^3}{\log n} \text{ and large } n
\end{aligned}
\tag{B.16}
$$

Resultantly, the contribution of subcase (i) of case (1) to the probability that both x and y are isolated can be upper-bounded as:

$$
\begin{aligned}
p_{21(i)} &\leq \pi r'^2(n)(1-p)(1 - q\frac{15\pi r^2(n)}{16})^{n-2} \\
&< \pi r^2(n)(1 - \frac{15q\pi r^2}{16}(n))^{n-2} \\
&\leq \pi r^2(n)(1 - \frac{5}{4}p\pi r^2(n))^{n-2} \quad (\because \frac{f}{c} \leq \frac{(\log \log n)^3}{\log n}) \\
&\leq \pi r^2(n)e^{-\frac{5}{4}(n-2)p\pi r^2(n)} \text{ from Lemma 48} \\
&\leq \left(\frac{2\alpha \log^2(n)}{n}\right) e^{-\frac{5}{4}(n-2)p\pi r^2(n)} \quad \text{from (B.13)} \\
&\leq e^{-\frac{5}{4}\log n - \frac{5}{4}b + \frac{5(\log n + b)}{2n} - \log n + \log 2\alpha + 2\log \log n} \\
&\leq e^{-\frac{17}{8}\log n - \frac{5}{4}b} \text{ for large } n \\
&\leq e^{-2\log n - b(n) - \frac{1}{2}\log \log \log n}
\end{aligned}
\tag{B.17}
$$

For sub-case (i) of case (1), when $\frac{f}{c} > \frac{(\log \log n)^3}{\log n}$, we lower bound the probability of a node being able to communicate with either of x and y by the probability that it is able to communicate with one of them (say x). Thus the probability that both x and y are isolated is at most:

$$p_{21(i)} \leq \pi r'^2(n)(1-p)(1-p\pi r^2(n))^{n-2}$$

$$\leq \left(\frac{256(\log\log n)^2}{\log^2 n} \right) \pi r^2(n)(1-p\pi r^2(n))^{n-2}$$

$$\leq \left(\frac{256(\log\log n)^2(\log n + b(n))}{pn\log^2 n} \right) (1-p\pi r^2(n))^{n-2}$$

$$\leq \left(\frac{256(\log\log n)^2\log n(\log n + b(n))}{n(\log\log n)^3\log^2 n} \right) (1-p\pi r^2(n))^{n-2}$$

$$(\because p \geq \frac{f}{c} > \frac{(\log\log n)^3}{\log n})$$

$$\leq \left(\frac{256(\log\log n)^2(2\log^2 n)}{n(\log\log n)^3\log^2 n} \right) (1-p\pi r^2(n))^{n-2}$$

$$\leq \left(\frac{512}{n\log\log n} \right) e^{-(n-2)p\pi r^2(n)} \text{ from Lemma 48}$$

$$\leq e^{-\log n - b(n) + \frac{2(\log n + b)}{n} - \log n + \log 512 - \log\log\log n}$$

$$\leq e^{-2\log n - b(n) - \frac{1}{2}\log\log\log n} \text{ for large } n$$

(B.18)

From (B.17) and (B.18), in sub-case (i), for all f, and large enough n:

$$p_{21(i)} \leq e^{-2\log n - b(n) - \frac{1}{2}\log\log\log n} \tag{B.19}$$

For sub-case (ii), the situation is depicted in Fig. B.3. The probability that some node can talk to either x or y is lower bounded by the probability that it lies in range of x (this probability is $\pi r^2(n)$) and shares a channel with it (the probability of sharing a channel is p), or it lies out of range of x but within range of y (at least $\frac{\sqrt{3}r(n)r'(n)}{2}$ for large enough n)[2], and shares a channel with y (once again this probability is p). The probability that

---

[2]The area within range of y but out of range of x is given by $\pi r^2(n) - $ overlap area ; where overlap area $= 2$ (area of quadrant subtending angle $2\theta - $ area of $\triangle ABC$) $\leq \pi r^2(n) - r^2(n)\sin(2\theta)$. Note that $\frac{\pi}{3} \leq \theta \leq \frac{\pi}{2}$ for sub-case (ii). Thus, the non-overlap area $\geq r^2(n)\sin(2\theta) = r^2(n)(2\sin\theta\cos\theta) = r^2(n)(2\sin\theta)(\frac{d(n)}{2r(n)}) \geq 2r^2(n)(\sin\frac{\pi}{3})(\frac{r'(n)}{2r(n)}) \geq \frac{\sqrt{3}r(n)r'(n)}{2}$.

both x and y are isolated can thus be upper bounded as:

$$
\begin{aligned}
p_{21(ii)} &\leq (\pi r^2(n) - \pi r'^2(n))(1-p)\left(1 - p\left(\pi r^2(n) + \frac{\sqrt{3}r(n)r'(n)}{2}\right)\right)^{n-2} \\
&\leq \pi r^2(n)\left(1 - p\left(\pi r^2(n) + \frac{\sqrt{3}r(n)r'(n)}{2}\right)\right)^{n-2} \\
&\leq \pi r^2(n)\left(1 - p\pi r^2(n)\left(1 + \frac{\sqrt{3}r'(n)}{2\pi r(n)}\right)\right)^{n-2} \\
&\leq \pi r^2(n)\left(1 - p\pi r^2(n)\left(1 + \frac{8\sqrt{3}\log\log n}{\pi \log n}\right)\right)^{n-2} \\
&\leq \pi r^2(n)e^{-(n-2)p\pi r^2(n)(1+\frac{4\log\log n}{\log n})} \quad \text{from Lemma 48} \quad (\because \pi < 2\sqrt{3})) \\
&\leq \left(\frac{2\alpha \log^2 n}{n}\right)e^{-(n-2)p\pi r^2(n)(1+\frac{4\log\log n}{\log n})} \quad \text{from (B.2)} \\
&\leq e^{-(n-2)p\pi r^2(n)(1+\frac{4\log\log n}{\log n})+\log 2\alpha + 2\log\log n - \log n} \\
&\leq e^{-\log n - b(n) - 4\log\log n + \frac{2(\log n + b(n))(1+\frac{4\log\log n}{\log n})}{n} + \log 2\alpha + 2\log\log n - \log n} \\
&\leq e^{-2\log n - b(n) - \log\log n} \text{ for large } n
\end{aligned}
\tag{B.20}
$$

For case 2, the probability that some node can communicate with either x or y is lower bounded by the probability that it lies in range of x (which is $\pi r^2(n)$) and shares a channel with it (which is p), or it lies out of range of x but within range of y (the disjunction of the two circles in Fig. B.1 (2) has area at least $\frac{1}{2}\pi r^2(n)$), and shares a channel with it. Thus the contribution of this case to the probability that both x and y are isolated is upper bounded by:

$$
\begin{aligned}
p_{22} &\leq \left(4\pi r^2(n) - \pi r^2(n)\right)\left(1 - p\pi r^2(n) - \frac{1}{2}p\pi r^2(n)\right)^{n-2} \\
&\leq \left(4\pi r^2(n) - \pi r^2(n)\right)\left(1 - \frac{3}{2}p\pi r^2(n)\right)^{n-2} \\
&\leq 3\pi r^2(n)e^{-\frac{3}{2}(n-2)p\pi r^2(n)} \text{ from Lemma 48} \\
&\leq \left(\frac{6\alpha \log^2 n}{n}\right)e^{-(n-2)\frac{3(\log n + b(n))}{2n}} \text{ from (B.13)} \\
&\leq e^{-\frac{3}{2}\log n - \frac{3}{2}b(n) + \frac{3(\log n + b(n))}{n} - \log n + \log 6\alpha + 2\log\log n} \\
&\leq e^{-\frac{9}{4}\log n - \frac{3}{2}b(n)} \text{ for large } n
\end{aligned}
\tag{B.21}
$$

261

The contribution of case 3 to the probability that both x and y are isolated is given by:

$$
\begin{aligned}
p_{23} &\leq (1 - 4\pi r^2)(1 - 2p\pi r^2(n))^{n-2} \\
&\leq (1 - 2p\pi r^2(n))^{n-2} \\
&\leq e^{-2(n-2)p\pi r^2(n)} \text{ from Lemma 48} \\
&\leq e^{-2\log n - 2b + \frac{2(\log n + b)}{n}}
\end{aligned}
\tag{B.22}
$$

Then, the probability p_2 that nodes i and j are both isolated is given by:

$$
p_2 = p_{21(i)} + p_{21(ii)} + p_{22} + p_{23}
\tag{B.23}
$$

Let us first consider the case where $b(n) = b$ is a constant.

$$
\begin{aligned}
\Pr[\text{ disconnection }] &\geq \sum_x \Pr[x \text{ is only isolated node}] \\
&\geq \sum_x \Pr[x \text{ isolated }] - \sum_{x,y} \Pr[x \text{ and } y \text{ both isolated }] \\
&= np_1 - n(n-1)p_2 \\
&\geq n(1 - p\pi r^2(n))^{(n-1)} - n(n-1)(p_{21(i)} + p_{21(ii)} + p_{22} + p_{23}) \\
&\geq \theta e^{-b} - n(n-1)\left(e^{-2\log n - b - \frac{1}{2}\log\log\log n} \right. \\
&\qquad \left. + e^{-2\log n - b - \log\log n} + e^{-\frac{9}{4}\log n - \frac{3}{2}b} + e^{-2\log n - 2b + \frac{2(\log n + b)}{n}} \right) \\
&\geq \theta e^{-b} - (1 + \epsilon)e^{-2b} \\
&\qquad \text{for any } \theta < 1, \epsilon > 0, \text{ and large } n \quad \text{(Lemma 48, Lemma 49)}
\end{aligned}
\tag{B.24}
$$

Now consider the case where $b(n)$ is not constant, and $\limsup\limits_{n\to\infty} b(n) = b$. Then, for any $\epsilon > 0$, $b(n) - b \leq \epsilon$ for large n. Since the probability of disconnection monotonically decreases in $b(n)$, we can take the following bound:

$$
\Pr[\text{ disconnection }] \geq \theta e^{-(b+\epsilon)} - (1 + \epsilon)e^{-2(b+\epsilon)}
\tag{B.25}
$$
$$
(\text{ for large enough } n)
$$

Thus, if $\limsup\limits_{n\to\infty} b(n) < +\infty$, the network is asymptotically disconnected with some positive probability.

Appendix C

Complete Proof of Scheduling Result (Theorem 13)

Recall the notation introduced in Chapter 5. Also recall that the arrival process at any link is i.i.d. over all time-slots, and that $E[\lambda_l(t)\lambda_k(t)]$ is bounded, i.e., $E[\lambda_l(t)\lambda_k(t)] \leq \eta$ for all $l \in \mathcal{L}, k \in \mathcal{L}$, where η is a suitable constant (hence $E[(\lambda_l(t))^2]$ is also upper-bounded by η). As mentioned in Chapter 5, we adopt the following convention: at the beginning of each time-slot, the scheduling decisions are taken, and transmissions occur. Then new arrivals occur at the end of the slot.

Let the queue-length of the queue for link l and channel c at the start of time-slot t be denoted by $q_l^c(t)$. Let the rate-allocated to link l in slot t over channel c be denoted by $x_l^c(t)$. Since we are considering single-interface nodes, at most one of the $x_l^c(t)$'s is non-zero for a link l. Furthermore $x_l^c(t) = 0$ if link l is not scheduled over channel c in slot t, and $x_l^c(t) = r_l^c$ else.

Recall that $r_l = \max_{c \in \mathcal{C}} r_l^c$. From the assumptions stated in Chapter 5, $r_l^c > 0$ for all $l \in \mathcal{L}, c \in \mathcal{C}$. Resultantly, $r_l > 0$ for all $l \in \mathcal{L}$.[1]

The queue dynamics are as follows:

$$q_l^c(t+1) = q_l^c(t) + \lambda_l^c(t) - x_l^c(t) \text{ where } \lambda_l^c(t) = \frac{\lambda_l(t)r_l^c}{\sum_{b \in \mathcal{C}} r_l^b} \tag{C.1}$$

We define the following Lyapunov function:

$$V_q(\overrightarrow{q}(t)) = \sum_{l \in \mathcal{L}} \sum_{c \in \mathcal{C}} \left[\frac{q_l^c(t)}{r_l^c} \left(\sum_{k \in \mathcal{A}(l)} \sum_{d \in \mathcal{C}} \frac{q_k^d(t)}{r_k^d} + \sum_{k \in \mathbf{I}'(l)} \frac{q_k^c(t)}{r_k^c} \right) \right] \tag{C.2}$$

This Lyapunov function is somewhat similar in form to that used in [120].

---

[1] As also stated in Chapter 5, the results can be easily generalized to the case when $r_l^c = 0$ for some l, c. However, even in those scenarios, it is reasonable to assume that $r_l > 0$ for all $l \in \mathcal{L}$, since any feasible load-vector must have $\lambda_l = 0$ for any link l with $r_l = 0$, and such links can be ignored/eliminated from consideration beforehand.

It can be seen that:

$$V_q(\overrightarrow{q}(t+1)) - V_q(\overrightarrow{q}(t)) = \sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{q_l^c(t+1)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t+1)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t+1)}{r_k^c}\right)\right]$$

$$-\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t)}{r_k^c}\right)\right]$$

$$=\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{(q_l^c(t)+q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t)+q_k^d(t+1)-q_k^d(t))}{r_k^d}\right.\right.$$

$$\left.\left.+\sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t)+q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right] - \sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t)}{r_k^c}\right)\right]$$

$$=\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left[\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t)}{r_k^c}\right)\right.$$

$$\left.+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t+1)-q_k^d(t))}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{(q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t)}{r_k^c}\right)\right]$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{(q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t+1)-q_k^d(t))}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

$$-\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t)}{r_k^c}\right)$$

$$=\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t+1)-q_k^d(t))}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{(q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t)}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{q_k^c(t)}{r_k^c}\right)\right]$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{(q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t+1)-q_k^d(t))}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

$$=2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t+1)-q_k^d(t))}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\left[\frac{(q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{(q_k^d(t+1)-q_k^d(t))}{r_k^d} + \sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

since $k \in \mathcal{A}(l) \implies l \in \mathcal{A}(k)$ and $k \in \mathbf{I}'(l) \implies l \in \mathbf{I}'(k)$ from the symmetric conflicts assumption

(C.3)

Denote by $\mathcal{L}'(t)$ the set of link-channel pairs (l, c) for which $q_l^c(t) \geq r_l^c$. This set of link-channel pairs participates in the scheduling process for slot t. By design, the scheduler computes a maximal schedule over all participating links. Therefore, for all $(l, c) \in \mathcal{L}'(t)$:

$$\sum_{k \in \mathcal{A}(l)} \sum_{d \in \mathcal{C}} \frac{x_k^d(t)}{r_k^d} + \sum_{k \in \mathbf{I}'(l)} \frac{x_k^c(t)}{r_k^c} \geq 1 \qquad \text{(C.4)}$$

If $\overrightarrow{\lambda}$ lies within the $\left(\frac{\sigma_s}{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}\right)$-reduced rate-region, then, by assumption, there exists some scheduling algorithm that achieves stability with load vector $\left(\frac{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}{\sigma_s}\right)\overrightarrow{\lambda}$. Similar to [74], we can argue that this implies existence of an average service-rate vector $\widetilde{x}_l^c$ for all l, c satisfying the following for some $\epsilon > 0$:

$$(1 + \epsilon)^2 \left(\frac{K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|}{\sigma_s}\right) \lambda_l \leq \sum_{c \in \mathcal{C}} \widetilde{x}_l^c \quad \text{for all links } l \qquad \text{(C.5)}$$

$$\sum_{k \in \mathbf{I}'(l)} \sum_{c \in \mathcal{C}} \frac{\widetilde{x_k^c}}{r_k^c} \leq K_{|\mathcal{C}|} \quad \text{for all links } l \qquad \text{(C.6)}$$

$$\sum_{k \in \mathcal{A}(l)} \sum_{d \in \mathcal{C}} \frac{\widetilde{x_k^d}}{r_k^d} \leq \max\{1, \gamma\} \quad \text{for all links } l \qquad \text{(C.7)}$$

Set $\overline{x_l^c} = \frac{\widetilde{x_l^c} \sigma_s}{(1+\epsilon)(K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|)}$. Then from (C.5), (C.6) and (C.7), we obtain that:

$$(1 + \epsilon)\lambda_l \leq \sum_{c \in \mathcal{C}} \overline{x_l^c} \quad \text{for all links } l \qquad \text{(C.8)}$$

$$\sum_{k \in \mathbf{I}'(l)} \sum_{c \in \mathcal{C}} \frac{\overline{x_k^c}}{r_k^c} \leq \frac{K_{|\mathcal{C}|} \sigma_s}{(1+\epsilon)(K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|)} \quad \text{for all links } l \qquad \text{(C.9)}$$

$$\sum_{k \in \mathcal{A}(l)} \sum_{d \in \mathcal{C}} \frac{\overline{x_k^d}}{r_k^d} \leq \frac{\max\{1, \gamma\} \sigma_s}{(1+\epsilon)(K_{|\mathcal{C}|} + \max\{1, \gamma\}|\mathcal{C}|)} \quad \text{for all links } l \qquad \text{(C.10)}$$

This yields that for all links l:

$$\sum_{b\in\mathcal{C}}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\overline{x_k^d}}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{\overline{x_k^b}}{r_k^b}\right)=\left(|\mathcal{C}|\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\overline{x_k^d}}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\sum_{b\in\mathcal{C}}\frac{\overline{x_k^b}}{r_k^b}\right)\qquad\text{(C.11)}$$

$$\leq\frac{\max\{1,\gamma\}\sigma_s|\mathcal{C}|}{(1+\epsilon)(K_{|\mathcal{C}|}+\max\{1,\gamma\}|\mathcal{C}|)}+\frac{K_{|\mathcal{C}|}\sigma_s}{(1+\epsilon)(K_{|\mathcal{C}|}+\max\{1,\gamma\}|\mathcal{C}|)}=\frac{\sigma_s}{1+\epsilon}<\sigma_s$$

Since $r_k^c\leq r_k$ for all channels c, therefore $\sum_{b\in\mathcal{C}}r_k^b\geq\sigma_s r_k\geq\sigma_s r_k^c$ for all $c\in\mathcal{C}$. Therefore, for all links l:

$$\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\sum_{b\in\mathcal{C}}\overline{x_k^b}}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\sum_{b\in\mathcal{C}}\overline{x_k^b}}{\sum_{b\in\mathcal{C}}r_k^b}\right)\leq\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\sum_{b\in\mathcal{C}}\frac{\overline{x_k^b}}{\sigma_s r_k}+\sum_{k\in\mathbf{I}'(l)}\sum_{b\in\mathcal{C}}\frac{\overline{x_k^b}}{\sigma_s r_k}\right)$$

$$\leq\frac{1}{\sigma_s}\sum_{b\in\mathcal{C}}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\overline{x_k^d}}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{\overline{x_k^b}}{r_k^b}\right)<1 \qquad\text{(C.12)}$$

$$\text{using (C.11))}$$

When $\lambda_l=0$ for all l, the queues are trivially stable. Hence, let us only consider the case where $\lambda_l>0$ for at least one link $l\in\mathcal{L}$. Let $y_{min}=\min\limits_{l\in\mathcal{L},\ \lambda_l>0}\frac{\lambda_l}{\sum_{b\in\mathcal{C}}r_l^b}$. Let $Q_{init}=\max\limits_{l\in\mathcal{L}}\frac{q_l^c(0)}{r_l^c}$, i.e., Q_{init} is the maximum of the initial queue-lengths. Note that if $\lambda_l=0$ for some link l, then $\frac{q_l^c(t)}{r_l^c}\leq\frac{q_l^c(0)}{r_l^c}\leq Q_{init}$.

Using (C.3):

$$E[V_q(\overrightarrow{q}(t+1))-V_q(\overrightarrow{q}(t))|\overrightarrow{q}(t)]$$

$$=2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}E[\frac{q_k^d(t+1)-q_k^d(t)}{r_k^d}]+\sum_{k\in\mathbf{I}'(l)}E[\frac{q_k^c(t+1)-q_k^c(t)}{r_k^c}]\right)$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}E\left[\frac{(q_l^c(t+1)-q_l^c(t))}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{q_k^d(t+1)-q_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{(q_k^c(t+1)-q_k^c(t))}{r_k^c}\right)\right]$$

$$\leq2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}E\left[\frac{\lambda_k^d(t)-x_k^d(t)}{r_k^d}\right]+\sum_{k\in\mathbf{I}'(l)}E\left[\frac{\lambda_k^c(t)-x_k^c(t)}{r_k^c}\right]\right)$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}E\left[\frac{\lambda_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k^c(t)}{r_k^c}\right)\right]$$

$$= 2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(E\left[\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k(t)}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k(t)}{\sum_{b\in\mathcal{C}}r_k^b}\right]-E\left[\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{x_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{x_k^c(t)}{r_k^c}\right]\right)$$

$$+\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}E\left[\frac{\lambda_l(t)}{\sum_{b\in\mathcal{C}}r_l^b}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k(t)}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k(t)}{\sum_{b\in\mathcal{C}}r_k^b}\right)\right]$$

$$\leq 2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(E\left[\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k(t)}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k(t)}{\sum_{b\in\mathcal{C}}r_k^b}\right]-E\left[\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{x_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{x_k^c(t)}{r_k^c}\right]\right)+C_1$$

$$= 2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}-E\left[\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{x_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{x_k^c(t)}{r_k^c}\right]\right)+C_1$$

$$= 2\sum_{(l,c)\in\mathcal{L}'(t)}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}-E\left[\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{x_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{x_k^c(t)}{r_k^c}\right)\right]\right)$$

$$+2\sum_{(l,c)\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}-E\left[\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{x_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{x_k^c(t)}{r_k^c}\right)\right]\right)+C_1$$

$$\leq 2\sum_{(l,c)\in\mathcal{L}'(t)}\frac{q_l^c(t)}{r_l^c}\left[\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}\right)-\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\sum_{b\in\mathcal{C}}\overline{x_k^b}}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\sum_{b\in\mathcal{C}}\overline{x_k^b}}{\sum_{b\in\mathcal{C}}r_k^b}\right)\right.$$

$$+\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\sum_{b\in\mathcal{C}}\overline{x_k^b}}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\sum_{b\in\mathcal{C}}\overline{x_k^b}}{\sum_{b\in\mathcal{C}}r_k^b}\right)-E\left[\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{x_k^d(t)}{r_k^d}+\sum_{k\in\mathbf{I}'(l)}\frac{x_k^c(t)}{r_k^c}\right)\right]$$

$$\left.+2\sum_{l\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}\right)\right]+C_1$$

$$\leq 2\sum_{(l,c)\in\mathcal{L}'(t)}\frac{q_l^c(t)}{r_l^c}\left[-\epsilon\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}\right)\right]$$

$$+2\sum_{l\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)}\frac{q_l^c(t)}{r_l^c}\left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}+\sum_{k\in\mathbf{I}'(l)}\frac{\lambda_k}{\sum_{b\in\mathcal{C}}r_k^b}\right)+C_1$$

using (C.8), (C.4) and (C.12)

$$\leq 2 \sum_{(l,c)\in\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c} \left[-\epsilon \left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}} \frac{\lambda_k}{\sum_{b\in\mathcal{C}} r_k^b} + \sum_{k\in\mathbf{I}'(l)} \frac{\lambda_k}{\sum_{b\in\mathcal{C}} r_k^b} \right) \right] - 2 \sum_{l\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c}\epsilon y_{min}$$

$$+ 2 \sum_{(l,c)\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c}\epsilon y_{min} + 2 \sum_{l\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c} \left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}} \frac{\lambda_k}{\sum_{b\in\mathcal{C}} r_k^b} + \sum_{k\in\mathbf{I}'(l)} \frac{\lambda_k}{\sum_{b\in\mathcal{C}} r_k^b} \right) + C_1$$

(subtracting and adding back $2 \displaystyle\sum_{l\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c}\epsilon y_{min}$)

$$\leq 2\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}} \frac{q_l^c(t)}{r_l^c}(-\epsilon y_{min}) + 2\epsilon y_{min} \sum_{\substack{l\in\mathcal{L} \\ \lambda_l=0}}\sum_{c\in\mathcal{C}} Q_{init} + 2\epsilon y_{min} \sum_{(l,c)\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c}$$

$$+ 2 \sum_{(l,c)\in(\mathcal{L}\times\mathcal{C})-\mathcal{L}'(t)} \frac{q_l^c(t)}{r_l^c} \left(\sum_{k\in\mathcal{A}(l)}\sum_{d\in\mathcal{C}} \frac{\lambda_k}{\sum_{b\in\mathcal{C}} r_k^b} + \sum_{k\in\mathbf{I}'(l)} \frac{\lambda_k}{\sum_{b\in\mathcal{C}} r_k^b} \right) + C_1$$

$$\leq -2\epsilon\frac{y_{min}}{r_{max}}\sum_{l\in\mathcal{L}}\sum_{c\in\mathcal{C}} q_l^c(t) + C_3$$

where $r_{max} = \max_{l\in\mathcal{L},c\in\mathcal{C}} r_l^c$, $C_1 = \frac{|\mathcal{L}||\mathcal{C}|\eta(A_{max}|\mathcal{C}|+I_{max})}{(\min_{l\in\mathcal{L}} r_l)^2}$, and $C_3 = C_1 + 2\epsilon y_{min}|\mathcal{L}||\mathcal{C}|Q_{init} + 2\epsilon y_{min}|\mathcal{L}||\mathcal{C}| + 2|\mathcal{L}||\mathcal{C}|(A_{max}|\mathcal{C}| + I_{max})$.

Invoking Lemma 2 from [85], this proves stability.

Appendix D

Auxiliary Results Used in Broadcast Proofs

D.1 Justification for Approximate Argument used in Section 8.6

We claimed in Section 8.6 of Chapter 8 that, given a simple closed region S of area A, and perimeter p, bounded by up to k straight line segments and circular arcs of radius r, where k is a small constant, the number of lattice points in S is $A \pm O(p)$. We justify this by bounding S, within and without, by lattice polygons, and applying Pick's Theorem [113]. For any such region S, consider the lattice polygon comprising grid squares that lie completely within S (Fig. D.1). In certain cases, instead of a single lattice polygon, we obtain a number of simple polygons that may share a common vertex, or are disconnected (if S has narrow constrictions or *necks* (Fig. D.2)). In rare instances, no such polygon may be obtained, if S is extremely narrow, and has no grid square lying completely within it ($A = O(p)$ for such regions, and these can be ignored). We call the polygon(s) thus obtained P_{in} (in case of multiple polygons, P_{in} refers to their union). Note that $S - P_{in}$ comprises the grid squares that are partially in S, i.e., those traversed by the boundary of S. Since the boundary of S comprises up to k line segments and arcs of radius r, the number of grid squares traversed by the boundary is at most $2p + ck$, where c is a constant. The area of P_{in} must thus be at least $A - (2p + ck)$. Let n_1 denote the number of lattice points falling in P_{in}. Similarly, consider the lattice polygon P_{out} obtained by taking the union of all grid squares that lie fully or partially in S. P_{out} is simple, fully contains S, and its area can be no more than $A + (2p + ck)$ (it can at most have an additional area comprising the grid squares traversed by the boundary of S). Let the number of lattice points falling in P_{out} be n_2. Then $n_1 \leq N_l \leq n_2$. By invoking Pick's Theorem [1], it can be

---

[1]Pick's Theorem: Let A be the area of a simple closed lattice polygon. Let B denote the number of lattice points on the polygon boundary, and I the number of points in the polygon interior. Then: $A = I + \frac{1}{2}B - 1$.

shown that $n_1 \geq A - O(p)$, and $n_2 \leq A + O(p)$. Thus $N_l = A \pm O(p)$.
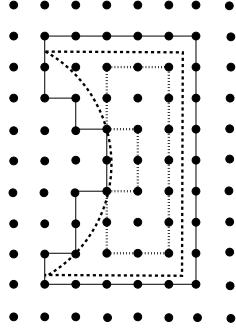

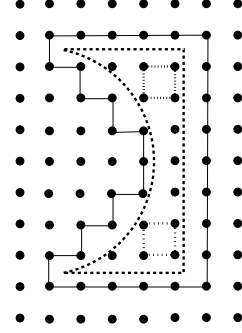Figure D.1: Bounding a Simple Closed Region via Lattice Polygons

Figure D.2: Region with *Neck*: Multiple Simple Polygons in Interior

D.2 Calculation of Collective Area of Regions A and B_1 from Section 8.6.

Consider Fig. D.3. Denote the regions within distance r of nodes N and M by $nbd(N)$ and $nbd(M)$ respectively. Then the collective area of regions A and $B_1 =$ Area of $nbd(N) \cap nbd(M)$ - Area of Sector HMJ + Area of $\triangle HMJ$. We show the calculations below. All angles are in radians. Sector KMR (HMJ) or $\triangle$ KMR (HMJ) refers to the sector/triangle subtending obtuse (and not reflex) angle KMR (HMJ) at M.

1. Area of $nbd(N) \cap nbd(M) = 2$ (Area of Sector KMR - Area of $\triangle$ KMR).

   Area of Sector KMR $= \pi r^2 \frac{\angle KMR}{2\pi} = \pi r^2 \frac{(2\cos^{-1}(\frac{r+1}{4r})))}{2\pi} \approx (r^2(\cos^{-1}(\frac{1}{4}))) \approx 1.318r^2$ for sufficiently large r.

   Area of $\triangle$ KMR $= \frac{1}{2}r^2 \sin(\angle KMR) \approx 0.242r^2$.

   Thus, Area of $nbd(N) \cap nbd(M) = 2(1.318 - 0.242)r^2 = 2(1.076)r^2 = 2.152r^2$.

2. Area of $\triangle HMJ = \frac{1}{2}r^2 sin(\angle HMJ) = \frac{1}{2}r^2 \sin(2\cos^{-1}(\frac{r+1}{2r})) \approx 0.433r^2$.

3. Area of Sector HMJ $= \pi r^2 \cdot \frac{\angle HMJ}{2\pi} = 1.047r^2$.

Thus collective area of A and B_1 is give by:

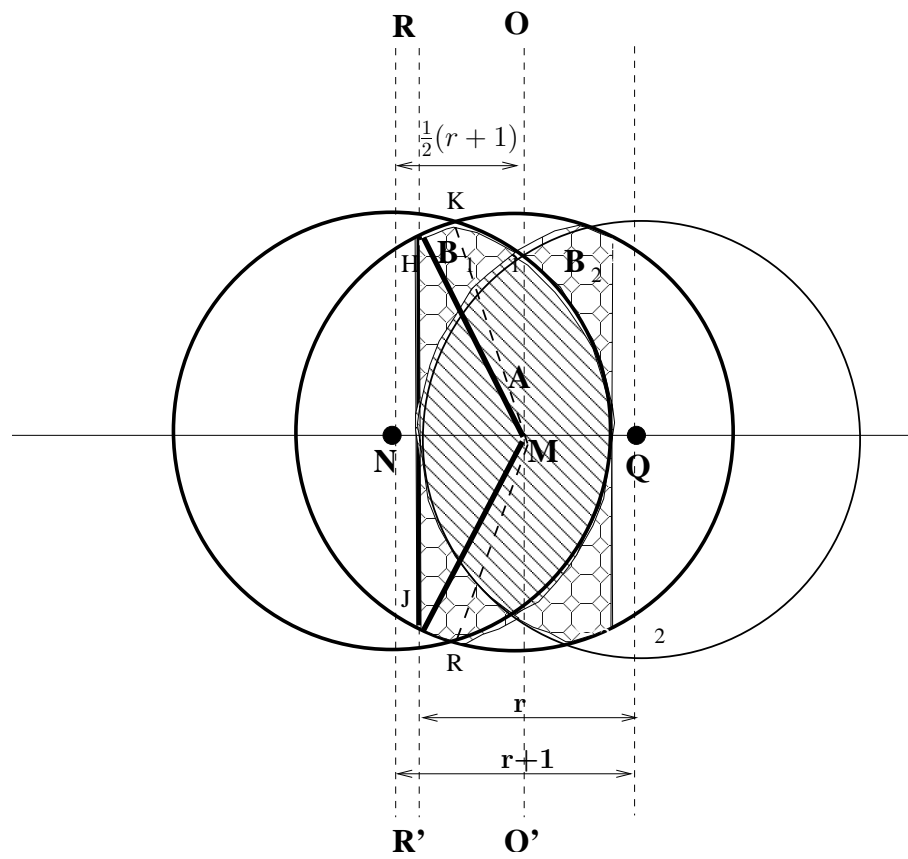 $2.152r^2 - 1.047r^2 + 0.433r^2 = 1.538r^2 \approx 0.49\pi r^2$.

Figure D.3: Calculation of Collective Area of Regions A and B_1 (from Fig. 8.6)

Appendix E

Useful Mathematical Results

In this appendix, we state some results that have been used in many of our proofs. Many of these are well-known results.

Fact 1. *For all $0 \leq x < 1$:*

$$\ln \frac{1}{1-x} \geq x$$

Fact 2. *For all $0 \leq x \leq 1$:*

$$(1-x) \leq e^{-x}$$

Lemma 50. *(Jogdeo & Samuels [47]) Given $X = Y_1 + Y_2 + ..., +Y_n$ where $\forall i, Y_i = Bernoulli(p_i)$, and $\sum p_i = np$, the median m of the distribution is either $\lfloor np \rfloor$ or $\lceil np \rceil$, i.e., $\Pr[X \leq m] \geq \frac{1}{2}$ and $\Pr[X \geq m] \geq \frac{1}{2}$.*

Lemma 51. *(Chernoff Bound [83]) Let $X_1, ..., X_n$ be independent Poisson trials, where $\Pr[X_i = 1] = p_i$. Let $X = \sum\limits_{i=1}^{n} X_i$. Then, for any $\beta > 0$:*

$$\Pr[X \geq (1+\beta)E[X]] \leq \left(\frac{e^{\beta}}{(1+\beta)^{(1+\beta)}} \right)^{E[X]} \tag{E.1}$$

Lemma 52. *(Chernoff Upper Tail Bound [83]) Let $X_1, ..., X_n$ be independent Poisson trials, where $\Pr[X_i = 1] = p_i$. Let $X = \sum\limits_{i=1}^{n} X_i$. Then, for $0 < \beta \leq 1$:*

$$\Pr[X \geq (1+\beta)E[X]] \leq \exp(-\frac{\beta^2}{3}E[X]) \tag{E.2}$$

Lemma 53. *(Chernoff Lower Tail Bound [83]) If $X = \sum\limits_{i=1}^{n} X_i$, where each X_i is independent and $Bernoulli(p_i)$, then for $0 < \beta < 1$:*

$$\Pr[X \leq (1-\beta)E[X]] \leq \exp(-\frac{\beta^2}{2}E[X]) \tag{E.3}$$

Lemma 54. *(Relative Entropy Form of Chernoff-Hoeffding Bound[45]) If* $X = \sum_{i=1}^{n} X_i$, *where each* X_i *is Bernoulli(p), then for* $p \leq \beta \leq 1$:

$$\Pr[X \geq \beta n] \leq e^{-n(\beta \ln \frac{\beta}{p} + (1-\beta) \ln \frac{1-\beta}{1-p})} \qquad \text{(E.4)}$$

Lemma 55. *The chernoff bounds continue to apply if the Poisson trials are not independent, but are negatively correlated.*

This is a well-known, and often-used result. See [87, 30]. Also see the proof for the Chernoff bound in [83], from which it can be seen that this holds.

Lemma 56. *[24] If* X_1, X_2,..., X_n *are drawn i.i.d. from alphabet* χ *according to* $Q(x)$, *then probability of sequence* $\mathbf{x}$ *is given by:*

$$Q^{(n)}(\mathbf{x}) = e^{-n(H(P_x) + D(P_\mathbf{x}||Q))} \qquad \text{(E.5)}$$

where H *and* P *denote the entropy and relative entropy functions (here considered w.r.t base e).*

Also, for any distributions P *and* Q, *the size of type class* $T(P)$ *satisfies:*

$$\frac{1}{(n+1)^{|\chi|}} e^{nH(P)} \leq |T(p)| \leq e^{nH(P)} \qquad \text{(E.6)}$$

and, the probability of the type class $T(P)$ *under* Q *is governed by:*

$$\frac{1}{(n+1)^{|\chi|}} e^{-n(D(P||Q))} \leq Q^{(n)}(T(p)) \leq e^{-n(D(P||Q))} \qquad \text{(E.7)}$$

Lemma 57. *(Vapnik-Chervonenkis Theorem) Let* S *be a set with finite VC dimension* $VCdim(\mathcal{S})$. *Let* $\{X_i\}$ *be i.i.d. random variables with distribution* P. *Then for* $\epsilon, \delta > 0$:

$$\Pr\left(\sup_{D \in \mathcal{S}} \left| \frac{1}{N} \sum_{i=1}^{N} I_{X_i \in D} - P(D) \right| \leq \epsilon \right) > 1 - \delta$$

$$\text{whenever } N > \max\left(\frac{8VCdim(\mathcal{S})}{\epsilon} \log_2 \frac{16e}{\epsilon}, \frac{4}{\epsilon} \log_2 \frac{2}{\delta} \right)$$

Lemma 58. *Suppose we are given a region of area n, with n nodes located uniformly at random. Consider all axis-parallel rectangles of area $a(n)$. If $a(n) = 100\alpha \log n, 1 \leq \alpha \leq \frac{n}{100 \log n}$, then each such rectangle has at least $100\alpha \ln n - 50 \log n$ nodes, with probability at least $1 - \frac{50 \ln n}{n}$.*

Proof. It is known that the set of axis-parallel rectangles in $\mathbb{R}^2$ has VC-dimension 4. We consider the set of all axis-parallel rectangles $\mathcal{S}$ of area $100\alpha \ln n$. Then considering the n random variables X_i denoting node positions, $\Pr[X_i \in D(D \in \mathcal{S})] = \frac{100\alpha \ln n}{n}$. Then, from the VC-theorem (Lemma 57):

$$\Pr\left(\sup_{D \in \mathcal{S}} \left| \frac{\text{No. of nodes in } D}{n} - \frac{100\alpha \ln n}{n} \right| \leq \epsilon(n) \right) > 1 - \delta(n)$$
$$\text{whenever } n > \max\left(\frac{32}{\epsilon} \log_2 \frac{16e}{\epsilon}, \frac{4}{\epsilon} \log_2 \frac{2}{\delta} \right)$$

This is satisfied when $\epsilon(n) = \delta(n) = \frac{50 \ln n}{n}$. Thus, with probability at least $1 - \frac{50 \ln n}{n}$, the population $\text{Pop}(D)$ of cell D satisfies:

$$100\alpha \ln n - 50 \ln n \leq \text{Pop}(D) \leq 100\alpha \ln n + 50 \ln n \tag{E.8}$$

This completes the proof. $\qquad\square$

Fact 3. *If we attempt to divide a $\sqrt{n} \times \sqrt{n}$ grid into disjoint neighborhoods in the L_∞ metric (as in Fig. 9.1), then the number of such disjoint neighborhoods that can be obtained is at least $\frac{(\lfloor \sqrt{n} \rfloor)^2}{(2r+1)^2} \geq \frac{(\sqrt{n}-1)^2}{4r^2+4r+1}$ for large n. Observing that $d = 4r^2 + 4r$ and $d \geq d_{min} = 8$, the number of such disjoint neighborhoods obtainable is at least $\frac{(\lfloor \sqrt{n} \rfloor)^2}{(2r+1)^2} \geq \frac{(\sqrt{n}-1)^2}{4r^2+4r+1} \geq \frac{n - 2\sqrt{n} + 1}{d(1 + \frac{1}{d})} \geq \frac{n}{2d}$ for large n, whenever r is such that $d = o(n)$.*

Lemma 59. *Suppose we are given a unit torus with n nodes located uniformly at random, and the region is sub-divided into axis-parallel square cells of area $a(n)$ each. If $a(n) = \frac{100\alpha(n) \log n}{n}, 1 \leq \alpha(n) \leq \frac{n}{100 \log n}$, then each cell has at least $(100\alpha(n) - 50) \log n$, and at most $(100\alpha(n) + 50) \log n$ nodes, with probability at least $1 - \frac{50 \log n}{n}$.*

Proof. It is known that the set of axis-parallel squares in $\mathbb{R}^2$ has VC-dimension 3. In our construction, we have a set of axis-parallel square cells $\mathcal{S}$ such that the cells all have area $a(n) = \frac{100\alpha \log n}{n}$. Then considering the n random variables X_i denoting node positions,

274

$\Pr[X_i \in D(D \in \mathcal{S})] = \frac{100\alpha \log n}{n}$. Then, from the VC-theorem (Lemma 57):

$$\Pr\left(\sup_{D \in \mathcal{S}} \left| \frac{\text{No. of nodes in } D}{n} - \frac{100\alpha(n) \log n}{n} \right| \leq \epsilon(n) \right) > 1 - \delta(n)$$

$$\text{whenever } n > \max\left(\frac{24}{\epsilon} \log_2 \frac{16e}{\epsilon}, \frac{4}{\epsilon} \log_2 \frac{2}{\delta} \right)$$

This is satisfied when $\epsilon(n) = \delta(n) = \frac{50 \log n}{n}$. Thus, with probability at least $1 - \frac{50 \log n}{n}$, the population $\text{Pop}(D)$ of cell D satisfies:

$$(100\alpha(n) - 50) \log n \leq \text{Pop}(D) \leq (100\alpha(n) + 50) \log n \tag{E.9}$$

$\square$

Lemma 60. *Suppose we are given a unit torus with n points(or nodes) located uniformly at random, let us consider the set of all circles of radius R and area $A(n) = \pi R^2$ on the unit torus. If $A(n) = \frac{100\alpha(n) \log n}{n}, 1 \leq \alpha(n) \leq \frac{n}{100 \log n}$, then each circle has at least $(100\alpha(n) - 50) \log n$, and at most $(100\alpha(n) + 50) \log n$ of these points (or nodes), w.h.p.*

Proof. The set of all circles of radius R in $\mathbb{R}^2$ has VC-dimension 3 (e.g., see [43]). Thereafter by the same argument as in the proof of Lemma 59, the result proceeds. $\square$

Lemma 61. *If n pairs of points (P_i, Q_i) are chosen uniformly at random in a unit area torus divided into square cells of area $a(n) = \Omega(\frac{\log n}{n})$, the resultant set of straight-line formed by each pair $L_i = P_i Q_i$ satisfies the condition that each cell has $O(n\sqrt{a(n)})$ lines passing through it w.h.p.*

Proof. Given the lines L_i are i.i.d., the proof argument of Lemma 3 in [36] can be applied to prove this result. $\square$

Lemma 62. *The number of subsets of size k chosen from a set of m elements is given by $\binom{m}{k} \leq \left(\frac{me}{k}\right)^k$.*

Theorem 31. *(Hall's Marriage Theorem [44], [92]) Given a set $\mathcal{S}$, let $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots \mathcal{T}_n\}$ be a finite system of subsets of $\mathcal{S}$. Then $\mathcal{T}$ possesses a system of distinct representatives if and only if for each k in $1, 2, .., n$, any selection of k of the sets $\mathcal{T}_i$ will contain between them at least k elements of $\mathcal{S}$. Alternatively stated: for all $\mathcal{A} \subseteq \mathcal{T}$, the following is true: $|\cup \mathcal{A}| \geq |\mathcal{A}|$*

Theorem 32. *(Integrality Theorem [22]) If the capacity function of a network flow graph takes on only integral values (i.e., each edge has integer capacity), then the maximum flow x produced by the Ford-Fulkerson method has the property that $|x|$ is integer-valued. Moreover, for all vertices u and v, the value of $x(u, v)$ is an integer.*

References

[1] Ashish Agarwal and P. R. Kumar. Improved capacity bounds for wireless networks. *Wirel. Commun. Mob. Comput.*, 4(3):251–261, 2004.

[2] Hagit Attiya and Jennifer Welch. *Distributed Computing*. McGraw-Hill, 1998.

[3] Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced Allocations. *SIAM J. Comput.*, 29(1):180–200, 2000.

[4] Paramvir Bahl, Ranveer Chandra, and John Dunagan. Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 ad-hoc wireless networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 216–230. ACM Press, 2004.

[5] Vartika Bhandari and Nitin H. Vaidya. On reliable broadcast in a radio network. In *PODC '05: Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 138–147. ACM Press, 2005.

[6] Vartika Bhandari and Nitin H. Vaidya. Capacity of multi-channel wireless networks with random (c, f) assignment. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 229–238. ACM Press, 2007.

[7] Vartika Bhandari and Nitin H. Vaidya. Connectivity and Capacity of Multichannel Wireless Networks with Channel Switching Constraints. In *Proceedings of IEEE INFOCOM*, pages 785–793, Anchorage, Alaska, May 2007.

[8] Vartika Bhandari and Nitin H. Vaidya. Heterogeneous multi-channel wireless networks: Scheduling and routing issues. Technical Report, UIUC, October 2007.

[9] Vartika Bhandari and Nitin H. Vaidya. Reliable Broadcast in Wireless Networks with Probabilistic Failures. In *Proceedings of IEEE INFOCOM*, pages 715–723, Anchorage, Alaska, May 2007.

[10] Vartika Bhandari and Nitin H. Vaidya. Reliable local broadcast in a wireless network prone to byzantine failures. In *Proc. of DIALM-POMC '07*, 2007.

[11] Vartika Bhandari and Nitin H. Vaidya. Secure capacity of multi-hop wireless networks with random key pre-distribution. In *Proceedings of IEEE Workshop on Mission Critical Networks*, 2008.

[12] Rishi Bhardwaj. Lessons from a multichannel wireless mesh network. M.S. Thesis, UIUC, 2007.

[13] Kenneth P. Birman, Mark Hayden, Oznur Oskasap, Zhen Xiao, Mihai Budiu, and Yaron Minsky. Bimodal multicast. *Transactions on Computer Systems (TOCS)*, 17(2):41–88, May 1999.

[14] Min Cao, Vivek Raghunathan, and P.R. Kumar. Cross layer exploitation of MAC layer diversity in wireless networks. In *Proc. of ICNP*, 2006.

[15] C. Carter, S. Yi, and R. Kravets. Arp considered harmful: Manycast transactions in ad hoc networks. In *Proc. of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2003.

[16] Antonio Caruso, Stefano Chessa, and Piero Maestrini. Worst-case diagnosis completeness in regular graphs under the pmc model. *IEEE Trans. Comput.*, 56(7):917–9249, 2007.

[17] Nicholas B. Chang and Mingyan Liu. Optimal channel probing and transmission scheduling for opportunistic spectrum access. In *MobiCom '07: Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 27–38. ACM, 2007.

[18] Chandrakanth Chereddi, Pradeep Kyasanur, and Nitin H. Vaidya. Design and implementation of a multi-channel multi-interface network. In *REALMAN '06: Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 23–30. ACM Press, 2006.

[19] Gregory Chockler, Murat Demirbas, Seth Gilbert, Nancy Lynch, Calvin Newport, and Tina Nolte. Reconciling the theory and practice of (un)reliable wireless broadcast. In *ICDCSW '05: Proceedings of the Fourth International Workshop on Assurance in Distributed Systems and Networks (ADSN) (ICDCSW'05)*, pages 42–48. IEEE Computer Society, 2005.

[20] Gregory Chockler, Murat Demirbas, Seth Gilbert, Calvin Newport, and Tina Nolte. Consensus and collision detectors in wireless ad hoc networks. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 197–206. ACM, 2005.

[21] Jeffrey Considine, Leonid A. Levin, and David Metcalf. Byzantine agreement with faulty majority using bounded broadcast. *CoRR*, cs.DC/0012024, 2000.

[22] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[23] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146. ACM, 2003.

[24] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., 1991.

[25] A. Dimakis and Jean Walrand. Sufficient conditions for stability of longest-queue-first scheduling: Second-order properties using fluid limits. *Advances in Applied Probability*, 38(2):505–521, 2006.

[26] Danny Dolev. The byzantine generals strike again. *J. Algorithms*, 3(1):14–30, 1982.

[27] Shlomi Dolev, Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Gossiping in a multi-channel radio network (an oblivious approach to coping with malicious interference). In *Proc. of DISC*, 2007.

[28] Olivier Dousse and Patrick Thiran. Connectivity vs capacity in dense ad hoc networks. In *Proceedings of IEEE INFOCOM*, 2004.

[29] Richard Draves, Jitendra Padhye, and Brian Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 114–128. ACM Press, 2004.

[30] Devdatt Dubhashi and Desh Ranjan. Balls and bins: a study in negative dependence. *Random Struct. Algorithms*, 13(2):99–124, 1998.

[31] R. Elz and R. Bush. Serial number arithmetic. RFC 1982.

[32] Stephen Fitzpatrick and Lambert Meertens. Experiments on dense graphs with a stochastic, peer-to-peer colorer. In *Workshop on Probabilistic Approaches in Search at AAAI'02*, 2002.

[33] C. Fragouli, J. Widmer, and J-Y Le Boudec. On the Benefits of Network Coding for Wireless Applications. In *4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006.

[34] M. Franceschetti, L. Booth, M. Cook, R. Meester, and J. Bruck. Percolation in wireless multi-hop networks. *Journal of Statistical Physics*, 2004.

[35] M. Franceschetti, O. Dousse, D. Tse, and P. Thiran. On the throughput capacity of random wireless networks. *IEEE Transactions on Information Theory*.

[36] Abbas El Gamal, James P. Mammen, Balaji Prabhakar, and Devavrat Shah. Throughput-Delay Trade-off in Wireless Networks. In *Proceedings of IEEE INFOCOM*, 2004.

[37] Leonidas Georgiadis, Michael J. Neely, and Leandros Tassiulas. Resource allocation and cross-layer control in wireless networks. *Found. Trends Netw.*, 1(1):1–144, 2006.

[38] Seth Gilbert, Rachid Guerraoui, and Calvin Newport. Of malicious motes and suspicious sensors. In *Proc. of OPODIS*, 2006.

[39] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1360–1369, 2001.

[40] S. Guha, K. Munagala, and S. Sarkar. Jointly optimal transmission and probing strategies for multichannel wireless systems. In *Proceedings of the 40th Annual Conference on Information Sciences and Systems*, pages 955–960, March 2006.

[41] Ramakrishna Gummadi, David Wetherall, Ben Greenstein, and Srinivasan Seshan. Understanding and mitigating the impact of rf interference on 802.11 networks. In *Proc. of ACM SIGCOMM '07*, 2007.

[42] Piyush Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In W. M. McEneany, G. Yin, and Q. Zhang, editors, *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, pages 547–566. Birkhauser, Boston, 1998.

[43] Piyush Gupta and P. R. Kumar. The Capacity of Wireless Networks. *IEEE Transactions on Information Theory*, IT-46(2):388–404, March 2000.

[44] P. Hall. On Representatives of Subsets. *J. London Math. Soc.*, s1-10(37):26–30, 1935.

[45] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[46] Information Sciences Institute. NS-2 network simulator. Version 2.31.

[47] Kumar Jogdeo and S. M. Samuels. Monotone convergence of binomial probabilities and a generalization of ramanujan's equation. *The Annals of Mathematical Statistics*, 39(4):1191–1195, August 1968.

[48] Changhee Joo. A local greedy scheduling scheme with provable performance guarantee. In *MobiHoc*, 2008.

[49] Changhee Joo, Xiaojun Lin, and Ness B. Shroff. Performance limits of greedy maximal matching in multi-hop wireless networks. In *CDC*, 2007.

[50] Changhee Joo, Xiaojun Lin, and Ness B. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, 2008.

[51] Changhee Joo and Ness B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. In *Proceedings of IEEE INFOCOM*, pages 19–27, 2007.

[52] Vikram Kanodia, Ashutosh Sabharwal, and Edward W. Knightly. MOAR: A Multi-Channel Opportunistic Auto-Rate Media Access Protocol for Ad Hoc Networks. In *BROADNETS*, pages 600–610, 2004.

[53] Bruno Kauffmann, François Baccelli, Augustin Chaintreau, Vivek Mhatre, Konstantina Papagiannaki, and Christophe Diot. Measurement-based self organization of interfering 802.11 wireless access networks. In *INFOCOM*, pages 1451–1459, 2007.

[54] V. Kawadia and P. R. Kumar. A cautionary perspective on cross-layer design. *IEEE Wireless Communications*, 12(1):3–11, 2005.

[55] D. Kim, C. Hsin, and M. Liu. Asymptotic connectivity of low duty-cycled wireless sensor networks. In *Proc. MILCOM*, 2005.

[56] S. Knappe, L. Liew, V. Shah, P. Schwindt, J. Moreland, L. Hollberg, and J. Kitching. A microfabricated atomic clock. *Appl. Phys. Lett.*, 85, 2004.

[57] Chiu-Yuen Koo. Broadcast in radio networks tolerating byzantine adversarial behavior. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 275–282. ACM Press, 2004.

[58] Chiu-Yuen Koo, Vartika Bhandari, Jonathan Katz, and Nitin H. Vaidya. Reliable broadcast in radio networks: The bounded collision case. In *Proceedings of ACM PODC 2006*, 2006.

[59] Ulas C. Kozat and Leandros Tassiulas. Throughput capacity of random ad hoc networks with infrastructure support. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 55–65. ACM Press, 2003.

[60] Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. Fault-tolerant broadcasting in radio networks. *J. Algorithms*, 39(1):47–67, 2001.

[61] S. Krishnamurthy, R. Chandrasekaran, N. Mittal, and S. Venkatesan. Brief announcement: Algorithms for node discovery and configuration in cognitive radio networks. In *Proc. of DISC '06 (LNCS 4167)*, pages 572–574. Springer-Verlag, 2006.

[62] Santosh Kumar, Ten H. Lai, and József Balogh. On k-coverage in a mostly sleeping sensor network. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 144–158, New York, NY, USA, 2004. ACM Press.

[63] Pradeep Kyasanur. Multi-channel wireless networks: Capacity and protocols. Ph.D. Thesis, UIUC, 2006.

[64] Pradeep Kyasanur, Chandrakanth Chereddi, and Nitin H. Vaidya. Net-x: System extensions for supporting multiple channels, multiple interfaces, and other interface capabilities. Technical Report, CSL, UIUC, August 2006.

[65] Pradeep Kyasanur and Nitin H. Vaidya. Capacity of Multi-channel Wireless Networks: Impact of Number of Channels and Interfaces. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 43–57. ACM Press, 2005.

[66] Pradeep Kyasanur and Nitin H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. Technical Report, CSL, UIUC, October 2006.

[67] Pradeep Kyasanur and Nitin H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(1):31–43, 2006.

[68] C. C. Lamb, L.S. DeBrunner, A. Das, and K. Thulasiraman. Distributed diagnosis for multiprocessor systems using extended local neighborhoods. In *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, 2000*, pages 384–387, 2000.

[69] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.

[70] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[71] Kuo-Chun Lee and Victor O. K. Li. A wavelength-convertible optical network. *Journal of Lightwave Technology*, 11, 1993.

[72] D.J. Leith and P. Clifford. Convergence of distributed learning algorithms for optimal wireless channel allocation. In *Proc. of 45th IEEE Conference on Decision and Control (CDC)*, pages 2980–2985, Dec. 2006.

[73] Soung Chang Liew. Capacity assigment in non-switching multi-channel networks. Ph.D. thesis, EECS, MIT, 1988.

[74] X. Lin and S. Rasool. A Distributed Joint Channel-Assignment, Scheduling and Routing Algorithm for Multi-Channel Ad-hoc Wireless Networks. In *Proceedings of IEEE INFOCOM*, pages 1118–1126, May 2007.

[75] Xiaojun Lin and Ness B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *Proceedings of IEEE INFOCOM*, pages 1804–1814, 2005.

[76] Benyuan Liu, Zhen Liu, and Donald F. Towsley. On the capacity of hybrid wireless networks. In *Proceedings of IEEE INFOCOM*, 2003.

[77] Junning Liu, Dennis Goeckel, and Don Towsley. Bounds on the gain of network coding and broadcasting in wireless networks. In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.

[78] J. Luo, P.Th. Eugster, and J.-P. Hubaux. Route driven gossip: Probabilistic reliable multicast in ad hoc networks. In *Proc. of IEEE INFOCOM*, 2003.

[79] Ritesh Maheshwari, Himanshu Gupta, and Samir R. Das. Mutichannel mac protocols for wireless networks. In *Proceedings of IEEE SECON*, 2006.

[80] Paolo Minero Massimo Franceschetti, Marco D Migliore. The capacity of wireless networks: Information-theoretic and physical limits. In *Proceedings of the Forty-Fifth Annual Allerton Conference*, pages 729–735, September 2007.

[81] Simone Merlin, Nitin H. Vaidya, and Michele Zorzi. Resource allocation in multichannel multi-radio wireless networks. In *Proceedings of IEEE INFOCOM*, 2008.

[82] Arunesh Mishra, Vivek Shrivastava, Dheeraj Agrawal, Suman Banerjee, and Samrat Ganguly. Distributed channel management in uncoordinated wireless environments. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 170–181. ACM, 2006.

[83] Michael Mitzenmacher and Eli Upfal. *Probability and computing.* Cambridge University Press, 2005.

[84] Jeonghoon Mo, Hoi-Sheung Wilson So, and Jean Walrand. Comparison of multichannel mac protocols. *IEEE Transactions on Mobile Computing*, 7(1):50–65, 2008.

[85] Michael J. Neely, Eytan Modiano, and Charles E. Rohrs. Dynamic power allocation and routing for time varying wireless networks. In *Proceedings of IEEE INFOCOM*, 2003.

[86] Clark T.-C. Nguyen. Rf mems in wireless architectures. In *DAC '05: Proceedings of the 42nd annual conference on Design automation*, pages 416–420, 2005.

[87] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the chernoff-hoeffding bounds. *SIAM J. Comput.*, 26(2):350–368, 1997.

[88] Jay A. Patel, Haiyun Luo, and Indranil Gupta. A cross-layer architecture to exploit multi-channel diversity with a single transceiver. In *Proceedings of IEEE INFOCOM Minisymposium*, Anchorage, AK, USA, May 2007.

[89] M. Pease, R. Shostak, and L. Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, 1980.

[90] Andrzej Pelc and David Peleg. Broadcasting with locally bounded byzantine faults. *Information Processing Letters*, 93(3):109–115, Feb 2005.

[91] Andrzej Pelc and David Peleg. Feasibility and complexity of broadcasting with random transmission failures. In *PODC '05: Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 334–341, 2005.

[92] Hazel Perfect. The Mathematics of AGMs. *The Mathematical Gazette*, 53(383):13–19, February 1969.

[93] Dragan Petrovic, Kannan Ramchandran, and Jan Rabaey. Overcoming untuned radios in wireless networks with network coding. *IEEE Transactions on Information Theory*, 52(6):2649–2657, June 2006.

[94] F. P. Preparata, G. Metze, and R. T. Chien. On the connection assignment problem of diagnosable systems. *IEEE Transactions on Electronic Computers*, EC-16(6):848–854, 1967.

[95] Arjunan Rajeswaran and Rohit Negi. Capacity of power constrained ad-hoc networks. In *Proceedings of IEEE INFOCOM*, 2004.

[96] Krishna N. Ramachandran, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Milind M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proceedings of IEEE INFOCOM*, 2006.

[97] Eric Rozner, Yogita Mehta, Aditya Akella, and Lili Qiu. Traffic-aware channel assignment in wireless lans. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(2):43–44, 2007.

[98] A. Sahai, R. Tandra, S. M. Mishra, and N. Hoven. Fundamental design tradeoffs in cognitive radio systems. In *Proc. of ACM TAPAS*, 2006.

[99] Nachum Shacham and Peter J. B. King. Architectures and performance of multichannel multihop packet radio networks. *IEEE Journal on Selected Areas in Communications*, 5(6):1013 – 1025, July 1987.

[100] Sanjay Shakkottai, R. Srikant, and Ness Shroff. Unreliable sensor grids: Coverage, connectivity, and diameter. In *Proc. of Infocom 2003*, 2003.

[101] Sanjay Shakkottai, R. Srikant, and Ness Shroff. Unreliable sensor grids: Coverage, connectivity, and diameter. In *Proc. of Infocom 2003*, 2003.

[102] Sanjay Shakkottai, R. Srikant, and Ness Shroff. Correction to unreliable sensor grids: Coverage, connectivity, and diameter. Personal Communication, 2005.

[103] Gaurav Sharma, Ravi R. Mazumdar, and Ness B. Shroff. On the complexity of scheduling in wireless networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 227–238. ACM, 2006.

[104] Thomas Shen and Nitin H. Vaidya. Experiments on a multichannel multi-interface wireless mesh network. Technical Report, CSL, UIUC, 2008.

[105] Hoi-Sheung Wilson So, Giang Nguyen, and Jean Walrand. Practical synchronization techniques for multi-channel mac. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 134–145, New York, NY, USA, 2006. ACM Press.

[106] Jungmin So and Nitin H. Vaidya. Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pages 222–233. ACM Press, 2004.

[107] T. K. Srikanth and Sam Toueg. Optimal clock synchronization. *J. ACM*, 34(3):626–645, 1987.

[108] Suresh Subramaniam, Murat Azizolu, and Arun K. Somani. All-optical networks with sparse wavelength conversion. *IEEE/ACM Trans. Netw.*, 4(4):544–557, 1996.

[109] Lakshminarayanan Subramanian, Randy H. Katz, Volker Roth, Scott Shenker, and Ion Stoica. Reliable broadcast in unknown fixed-identity networks. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 342–351. ACM, 2005.

[110] Leandros Tassiulas and Anthony Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.

[111] Asimakis Tzamaloukas and J. J. Garcia-Luna-Aceves. Channel-hopping multiple access. In *ICC (1)*, pages 415–419, 2000.

[112] Vinod Vaikuntanathan. Brief announcement: broadcast in radio networks in the presence of byzantine adversaries. In *PODC '05: Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 167–167, 2005.

[113] Dale E. Varberg. Pick's theorem revisited. *The American Mathematical Monthly*, 92(8):584–587, October 1985.

[114] Ramanuja Vedantham, Sandeep Kakumanu, Sriram Lakshmanan, and Raghupathy Sivakumar. Component based channel assignment in single radio, multi-channel ad hoc networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 378–389. ACM, 2006.

[115] Evangelos Vergetis, Roch Guérin, and Saswati Sarkar. Realizing the benefits of user-level channel diversity. *SIGCOMM Comput. Commun. Rev.*, 35(5):15–28, 2005.

[116] Wei Wang, Xin Liu, and Dilip Krishnaswamy. Robust routing and scheduling in wireless mesh networks. In *Proc. of IEEE SECON*, June 2007.

[117] Ajit Warrier, Long Le, and Injong Rhee. Cross-layer optimization made practical. In *Proc. of Broadnets '07*, 2007.

[118] Haitao Wu, Fan Yang, Kun Tan, Jie Chen, Qian Zhang, and Zhensheng Zhang. Distributed Channel Assignment and Routing in Multi-radio Multi-channel Multi-hop Wireless Networks. *Journal on Selected Areas in Communications(JSAC)*, 24:1972–1983, November 2006.

[119] Xinzhou Wu and R. Srikant. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. In *Proceedings of IEEE INFOCOM*, 2006.

[120] Xinzhou Wu, R. Srikant, and James R. Perkins. Queue-length stability of maximal greedy schedules in wireless networks. In *Workshop on Information Theory and Applications*, 2006.

[121] Xinzhou Wu, R. Srikant, and James R. Perkins. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Trans. Mob. Comput.*, 6(6):595–605, 2007.

[122] Kaixin Xu, Mario Gerla, Lantao Qi, and Yantai Shu. Tcp unfairness in ad hoc wireless networks and a neighborhood red solution. *Wirel. Netw.*, 11(4):383–399, 2005.

[123] Feng Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.*, 10(2):169–181, 2004.

[124] Feng Xue and P. R. Kumar. On the theta-coverage and connectivity of large random networks. *Joint special issue of the IEEE Trans. on Information Theory and the IEEE/ACM Trans. on Networking on "Networking and Information Theory"*, May 2006.

[125] Feng Xue and P. R. Kumar. Scaling laws for ad hoc wireless networks. *Found. Trends Netw.*, 1(2):145–270, 2006.

[126] Jung Yee and Hossain Pezeshki-Esfahani. Understanding wireless lan performance trade-offs. *Communication Systems Design*, 2002.

[127] Wonyong Yoon, Jungmin So, and Nitin H. Vaidya. Routing exploiting multiple heterogeneous wireless interfaces: A tcp performance study. In *Proceedings of Military Communications Conference (MILCOM)*, 2006.

[128] Honghai Zhang and Jennifer C. Hou. Capacity of wireless ad-hoc networks under ultra wide band with power constraints. In *Proceedings of IEEE INFOCOM*, Miami, Florida, March 2005.

[129] Yi Zhao and Farokh B. Bastani. A self-adjusting algorithm for byzantine agreement. *Distrib. Comput.*, 5(4):219–226, 1992.

[130] Dong Zheng and Junshan Zhang. Protocol design and throughput analysis of opportunistic multi-channel medium access control. In *Proc. of CIIT'03*, November 2003.

Author's Biography

Vartika Bhandari was born in Allahabad, India in 1980. She received the B.Tech degree in Computer Science and Engineering from the Indian Institute of Technology Kanpur in 2002. In 2003, Vartika joined the Ph.D. program in Computer Science at the University of Illinois at Urbana-Champaign. She has been a member of the Wireless Networking Group in the Coordinated Science Laboratory since then, where she has worked on theoretical analysis and protocol design issues pertaining to wireless networks. One of the publications resulting from her Ph.D. research received the Best Student Paper Award at the the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc) in 2007. She is also a recipient of a Vodafone Graduate Fellowship for the years 2005-06, 2006-07 and 2007-08.